Visual Kit 5
Edição Standard/Professional

Versão 6.0.4.2

Guia do Usuário

TODOS OS DIREITOS RESERVADOS - EBENDINGER LTDA.

Conteúdo

Sobre este Manual....8
Ícones utilizados....8
I. Introdução....9
Um pouco de história....9
Edições do Visual Kit5....10
Edição Standard....10
Edição Professional....10
Edição Enterprise....10
Requisitos Técnicos do Visual Kit 5....10
Instalando o Visual Kit 5 no *Windows*....11
Razões para não ter conseguido fazer a instalação....15
Você consegue instalar mas não roda....16
II. Conceitos Básicos....19
Sistemas, módulos e entidades....19
Campos....22
Registros....23
Tabelas....24
Relacionamento....24
Um-para-Um....24
Um-para-Muitos....25
Índices e campos chave....26
Operações....27
III. O Ambiente de Desenvolvimento....28
Acessando o Visual Kit 5....28
Elementos do Ambiente....29
Janela Principal....29
Barra de Menus....30
Menu Arquivo....30
Menu Editar....30
Menu Ver....30
Menu Janela....31
Menu Help....31
Barra de Ferramentas....31
Menu *TreeView*....32
Módulo de Gerador de Sistemas....32
Módulo Ferramentas....34
Módulo Documentos....37
IV. Criando seu Primeiro Sistema Passo a Passo....38
A proposta....38
Desenhando o sistema....38
Carregando o Visual Kit 5....38

Gerando o diretório do sistema....39
Criação dos módulos e entidades....42
Desenhando o formulário....45
Inserindo os controles....46
Salvando o sistema e fechando o desenho....50
Acessando o sistema gerado....51
Testando o sistema....52
Incluindo dados....52
Consultando....55
Alterando dados....56
Excluindo dados....57
Modificando o sistema....57
V. Desenhando o Formulário....58
As ferramentas de desenho *Designer-Básica* e *Designer-Pro*....58
Manipulando controles....59
Conhecendo a Caixa de Ferramentas....59
Adicionando controles ao formulário....61
Selecionando controles....61
Posicionando controles....63
Dimensionando controles....64
Excluindo controles....64
Copiando, Recortando e Colando controles....64
Alinhamento de controles....65
Definindo ordem de tabulação....65
Entendendo as Propriedades....67
Manipulando a Janela de Propriedades....67
Coluna de Propriedades....68
Coluna de Valores....68
Área de Descrição....69
VI. A *Designer-Básica*....70
Principais controles....70
Form – Formulário....70
TextBox....72
ListBox....74
Exemplo de sistema com entidades relacionadas....75
ComboBox:....79
CheckBox e *OptionGroup*....81
Spinner....84
Imagem....84
Incluindo funcionalidades nos controles....85
Acessando dados pelo vetor *Vdados*....86
Áreas de relacionamento - acessando conteúdos de outras entidades....89
Funções de validações (pré e pós)....90
Exibe caixa de mensagem: Função *ferrmsg()*....90
Conteúdo automático - Função *altval()*....91

Retorna cálculos matemáticos....91
Somar dias a um campo data....91
Retorna o dia de uma data....91
Retorna o mês de uma data....91
Retorna o ano de uma data....91
Retorna hora do sistema....92
Retorna conteúdo de entidade relacionada....92
Coloca valor *default* - Função *rep_()*....92
Cria código seqüencial automático - Função *seq_()*....92
Impede campo vazio-Função *!empty()*....93
Limita o conteúdo do campo....93
Aceita S (Sim) ou N(Não)....93
Aceita valor num intervalo....93
Aceita um período de datas....93
Não permite conteúdo duplo - Função *dup_chk()*....94
Exibe conteúdo na tela – Função *prt ()*....94
Salta campo - Função *slt()*....95
Pesquisa se chave existe – Função *dbseek( )*....95
Busca última ocorrência da chave - Função *bprx( )*....95
Altera conteúdo de outras entidades - Função *chg_data ()*....96
Calcula o módulo 11 - Função *mod11( )*....96
Verifica hora - Função *validtime( )*....97
Altera variáveis do sistema - Função *altvar( )*....97
Alterar tamanho campo memo -Função *altmemo ()*....97
Seleciona área de relacionamento - Função *dbselect( )*....98
Retorna ao registro atual - Função *goatu( )*....98
Vai para o início do arquivo - Função *dbgotop( )*....98
Vai para o final do arquivo - Função *dbgobottom( )*....99
Movimenta o ponteiro do arquivo - Função *dbskip( )*....99
Grava dados em campo - Função *fieldput( )*....99
Exibe um controle objeto-imagem no formulário – Função *prt_image( )*....99
Criando validações compostas....100
Usando o operador AND....100
Usando o operador *iif()*....100
VII. A *Designer-Pro*: Novos Recursos....102
Principais controles da *Designer-Pro*....102
Novo *ListBox*....102
Exemplo de sistema com entidades relacionadas....103
Novo *ComboBox*:....109
Grid....112
Exemplo de entidade com chave composta....113
Botão....120
Exibe....120
Picture....120

Calendario....120
Link....121
Criando novas funcionalidades usando os eventos....121
Entendendo os eventos....121
Os principais eventos....123
O Editor de Expressões....124
Expressões matemáticas....124
O Editor de Macros....128
Funções bases cadastradas....132
Exibe janela com mensagem....133
Construir uma expressão como macro....133
Transforma em maiúsculas....134
Transforma em minúsculas....134
Operador condicional....135
Retorna uma quantidade específica de caracteres....136
Retira espaços em branco....137
Retorna dia da semana....138
Retorna o número do dia da semana de um campo data....138
Retorna o Valor Absoluto de um campo numérico....139
O Editor de Códigos....139
Utilizando o menu *Edit* para auxiliar a digitação de códigos....140
VIII. Geração de códigos fontes dos formulários....141
Gerando códigos em Visual Basic....141
Gerando códigos em Fox Pro....143
IX. Personalizando à apertura, o *about* e o manual 146
Imagem de apertura....146
O logo do *About*....147
O manual do sistema gerado....147
X. Criando um Sistema Completo pela *Designer-Básica*....148
A Proposta....148
Analisando o Sistema....148
Estrutura do Módulo 01 - Controle de Estoque....151
Estrutura do Módulo 02 – Contas a Pagar/Receber....159
Passos na Criação do Sistema....166
XI. Criando Sistemas Utilizando os Sistemas Brinde171
Gerando o sistema brinde....171
Alterando o sistema brinde....173
XII. Aproveitando dados....175
Acessando uma tabela .DBF existente....175
Acessando uma tabela .MDB existente....179
Importando dados de uma Planilha Excel (*.xls)....182

XIII. Instalando o Sistema Gerado em outro computador.....184
Instalando num único computador.....184
Instalando um Sistema Multiusuário.....186
XIV. Voltando ao Ambiente de Desenvolvimento....190
Renomeia e Copia Entidades.....190
XV. Criando Funções pelo FoxPro.....193
Criando funções específicas.....193
Regra do nome da função específica.....198
Cuidados na manipulação de arquivos.....199
Criando funções de validação de campos.....200
XVI. Criando um Sistema no MS SQL Server 2000.201
O SQL Server.....201
SQL Server- Visual Kit 5.....201
Criando sistemas com conexão remota via internet.....206
Rodando pelo navegador de internet.....206
XVII. Conhecendo as funcionalidades dos Sistema Gerados.....210
Localizar entidades e campos do sistema.....210
Criando usuários (*login* e *senha*) para o sistema.....211
Inclusão, alteração e exclusão de registros.....214
Consultas.....216
Relatórios rápidos, etiquetas e transferência.....218
Senhas dentro da entidade.....224
Indexação.....225
Outras opções úteis.....226
Filtro e ordenação.....226
Quebra.....228
Operações Matemáticas.....229
Consultas, Relatórios e Etiquetas Avançadas.....231
Origem dos Dados : preparando os dados.....231
Criando e personalizando Relatórios – *Report Designer*.....238
As bandas de um relatório.....239
Visualizando e Imprimindo.....240
Personalizando o Relatório.....240
Criando e personalizando Etiquetas – *Label Designer*.....244
Criando Códigos de Barras.....247
Instalar Fonte.....247
Criando o Código de Barra.....250
OLAP- Processamento Analítico *Online*.....255
O que é OLAP?.....255
Procedimentos para usar o OLAP.....256
Origem de Dados.....256

Criando o Cubo.....257
Navegando pelo Cubo.....259
Principais Definições.....260
Data Mining – Inteligência Artificial.....261
Definição.....261
Onde utilizar o Data Mining.....261
Recursos de Data Mining no Visual Kit 5.....262
Selecionando a Origem dos Dados.....263
Criando o Modelo.....264
Treinando e testando o modelo.....266
Usando o modelo.....267
XVIII. Apêndices.....268
A. Convertendo sistemas da versão 4.6 para a versão 6.0 do Visual Kit 5 268
B. Atualizando o Visual Kit 5 e sistemas gerados.....269
C. Denominação das Tabelas, Módulos, Entidades no Visual Kit 5.....271
D. Propriedade Formatos e Máscara de Edição.....272
E. Operadores.....275
Operadores de Caracteres.....275
Operadores de Datas.....275
Operadores Lógicos.....275
Operadores Numéricos.....275
Operadores Relacionais.....276
F. Regras de nomenclaturas Ebendinger.....277

Sobre este Manual

Este manual foi desenvolvido para os usuários do Visual Kit 5 Standard/Professional versão 6.0. É um guia tanto para pessoas sem conhecimentos profundos em programação e análise de sistemas como também para profissionais com experiência em desenvolvimento de sistemas comerciais que desejem aumentar sua produtividade utilizando esta ferramenta.

Você não precisa de experiência em programação para ler este manual de forma proveitosa e aprender com ele. Todos os recursos disponíveis são apresentados de forma clara, objetiva e didática sendo exemplificado para melhor compreensão do assunto.

O contexto técnico deste manual abrange todos os recursos básicos e intermediários disponíveis na ferramenta de desenvolvimento do Visual Kit 5.

Ícones utilizados

Os ícones são figuras para chamar sua atenção. Veja aqui o que os ícones utilizados neste livro significam.

Atenção! Mãos ao alto! – Trata-se de uma informação que você deve tentar se lembrar. Às vezes porque é uma informação útil; outras vezes, porque você pode ter problemas caso a esqueça.

Nota: Avisa o leitor que se trata de uma abordagem técnica que pode ser pulada, caso não seja de seu interesse.

Dica: Atalhos e dicas que podem poupar-lhe tempo e aborrecimentos.

Atenção! A primeira observação!
O Visual Kit 5 é um programa que constantemente insere novos recursos com o objetivo de satisfazer os usuários. Por isso, pode existir algum

recurso que você já tenha dentro do Visual Kit 5 e não esteja descrito aqui ou a contrário.

I. Introdução

Bem-vindo a este maravilhoso mundo de desenvolvimento de sistemas usando o Gerador de Sistemas Visual Kit 5!. Você está fazendo parte de uma grande e crescente comunidade de desenvolvedores que usam o Visual Kit 5 para criar os mais diversos sistemas.

O Visual Kit 5 oferece variados recursos que fazem dele um ambiente poderoso de desenvolvimento de sistemas dos últimos tempos.

Um pouco de história

O Visual Kit 5 é um produto da Ebendinger Systems Informática Ltda., empresa de informática fundada em 1989, que enfocou desde seu inicio o usuário não especializado em programação – a maioria no planeta-, e tratou seus sistemas como produtos, dando-lhes garantia de funcionamento e manutenção.

Em 1993, a Ebendinger lançou a primeira versão do produto o "*The Ebendinger Development Kit – Kit5 for DOS*". E em curtíssimo espaço de tempo tornou-se líder no mercado de Geradores de Sistemas. Em 1994, colocou no mercado a primeira versão de gerador de sistemas para Windows e Macintosh: o "*Visual Kit5"*, que a cada ano foi enriquecido em sucessivas versões. Formando parte deste processo, em 1997 abre uma filial nos EUA, com a finalidade de se expandir no mercado mundial.

Em 2001 lançou a versão 6.0, com uma nova cara, compatível com os novos sistemas operacionais gráficos com estrutura *tree-view* (em árvore). Também evoluiu para soluções que envolvem a geração de sistemas para ambiente via internet e com recursos de *OLAP* e *Data Mining*, em sintonia com as últimas tendências do mundo da informática. No 2002 o Visual Kit 5 alcançou um novo patamar, desdobrando-se em três versões: *Standard*, *Professional* e *Enterprise* orientadas para usuários não especializados, programadores e ambientes empresariais respectivamente.

A empresa é formada por duas divisões: A Divisão Industrial - para o desenvolvimento e comercialização de softwares e ferramentas que

ajudam no desenvolvimento de aplicativos e sistemas, e a Divisão de Consultoria para Sistemas Coorporativos, *Business Intelligence,* e Portais *Web*. Por último, existe uma outra área referida a edição de livros: a Ebendinger Editora.

Edições do Visual Kit5

Você poderá adquirir o Visual Kit 5 em três diferentes edições segundo suas necessidades e conhecimentos.

Edição Standard

O Visual Kit 5 *Standard* é uma excelente ferramenta de desenvolvimento de sistemas que não exige do usuário profundos conhecimentos de informática, programação ou análise de dados. Usando o Visual Kit 5, qualquer pessoa pode programar o sistema de seu interesse.

Os sistemas desenvolvidos são totalmente independentes e você poderá instalá-lo em outros computadores e comercializá-lo sem restrição nenhuma. Eles não precisarão do Visual Kit 5 para rodar.

Edição Professional

O Visual Kit 5 *Professional* possui todos os recursos da versão *Standard* e ainda a possibilidade de gerar código fonte dos formulários em várias nas linguagens de programação Fox Pro e Visual Basic.

Edição Enterprise

O Visual Kit 5 *Enterprise* é formado por 4 (quatro) ferramentas que o converte em um poderoso sistema de desenvolvimento permitindo aumentar a produtividade e a qualidade dos sistemas gerados por ele. Estas ferramentas são:

- Gerador automático de Sistemas;
- Gerador de formulários SCX e FRM;
- Gerador de formulários avançados;
- Gerador de Sistemas Visual Kit 5 .

Acesse o site da Ebendinger (www.ebendinger.com.br) para obter maiores informações sobre as edições do Visual Kit 5.

Requisitos Técnicos do Visual Kit 5

Computador/Processador	IBM PC ou 100% compatível.
Memória	Quantidade recomendada para o Windows, que você estiver utilizando.
Sistema Operacional	Windows 98 SE ou mais recente, com Internet Explorer 5.5 e Service Pack 2 ou mais recente Windows NT 4.5 com Service Pack 6 ou mais recente. Windows 2000 com Service Pack 2 ou mais recente. Você não pode utilizar o Visual Kit 5 versão 6.0 em outras configurações de Windows. As operações de OLAP e Data Mining estão disponíveis somente para o Windows 2000 com Service Pack 2, e SQL Server 2000 com Service Pack 1, ou superior. Não execute estas rotinas em outra configuração de Windows.
Disco Rígido	Ocupa aproximadamente 120 MB.
Unidade	Unidade de CD-ROM Obrigatória.
O Visual Kit 5 deverá ser executado preferencialmente com 1024 x 768 pontos de resolução, configurados para o seu Monitor. Não usar esta resolução de monitor, poderá impedir a visualização correta de todas as janelas do Visual Kit 5.	

Instalando o Visual Kit 5 no *Windows*

Instalar o Visual Kit 5 é muito simples, basta seguir as instruções do programa instalador. O Visual Kit 5 não atualiza os sistemas criados com um Visual Kit 5 de versão anterior. Os sistemas brinde não são instalados pelo Visual Kit 5.

Você tem espaço livre suficiente? Certifique-se de que possui espaço em disco suficiente para a instalação. Ela requer cerca de 120MB de espaço livre no seu disco rígido, não esqueça que após a instalação, você precisará de espaço para salvar os sistemas que criar. Portanto, a primeira coisa que precisa saber é se possui 120MB ou mais de espaço livre na unidade de disco rígido. Se não tiver, elimine alguns arquivos que não vá utilizar mais ou adquira uma nova unidade de disco rígido.

Atenção: O programa *Setup.exe* ou *WebApp.exe* não instala o Visual Kit 5 em computadores com menos de 32 MB de RAM ou com problemas no gerenciador de memória do Windows, ou na *cache* do seu micro. Nestes casos, antes de instalar aumente a quantidade de memória RAM do seu micro e/ou solucione os problemas, que existirem com o gerenciador de memória ou memória *cache*. Micros com problemas físicos na memória RAM ou memória *cache* não poderão executar o Visual Kit 5 corretamente

---

A continuação, descrevemos a instalação do Visual Kit 5 nos sistemas operacionais *Windows 98 SE, NT, 2000* ou *XP*:

- Insira o CD-ROM na unidade de CD-ROM

Se o programa de instalação não for executado pelo ***autorun*** do *Windows*, execute os passos abaixo:

- Clique o botão “Iniciar” (Start).
- Selecione a opção “Executar... (Run...)”
- Digite **d**:*setup.exe* tecle “ENTER” Se a unidade de CD-ROM está instalada no E:> substitua a letra **d**: por **e**:
- Siga as instruções do programa instalador. Observe as seguintes perguntas.

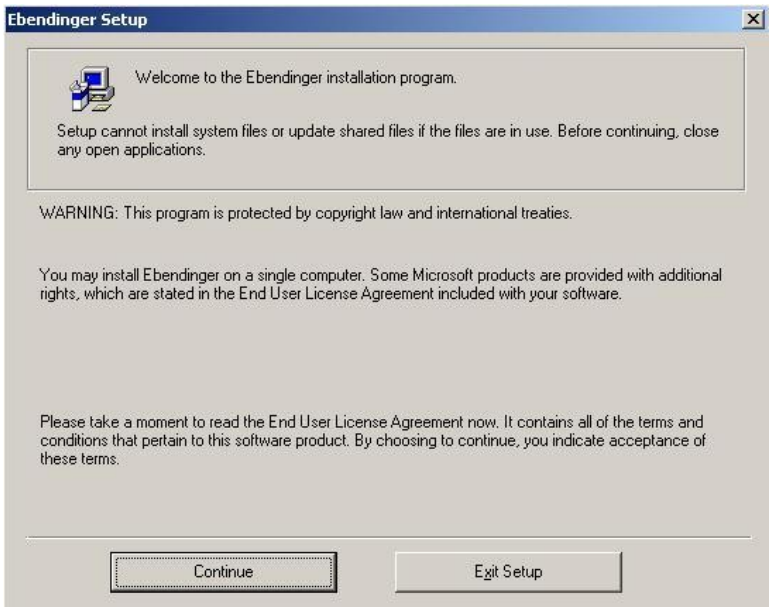


Fig 1 Clique no botão Continue**, para continuar a instalação.**

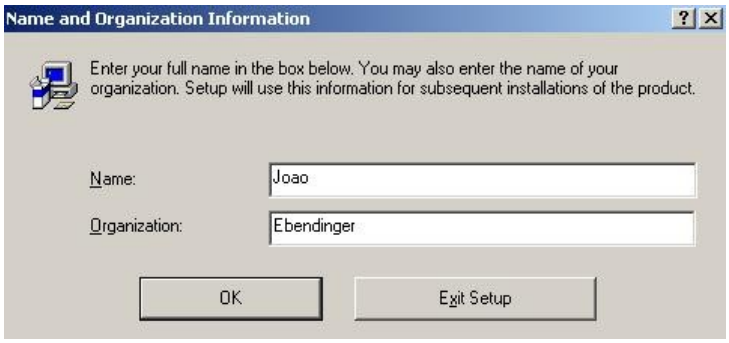


Fig 2 Cadastre o seu nome e o nome de sua empresa, nos campos acima indicados.

Fig 3 Para confirmar o seu nome e o nome de sua empresa, pressione o botão Ok**.**

Fig 4 Preparando instalação

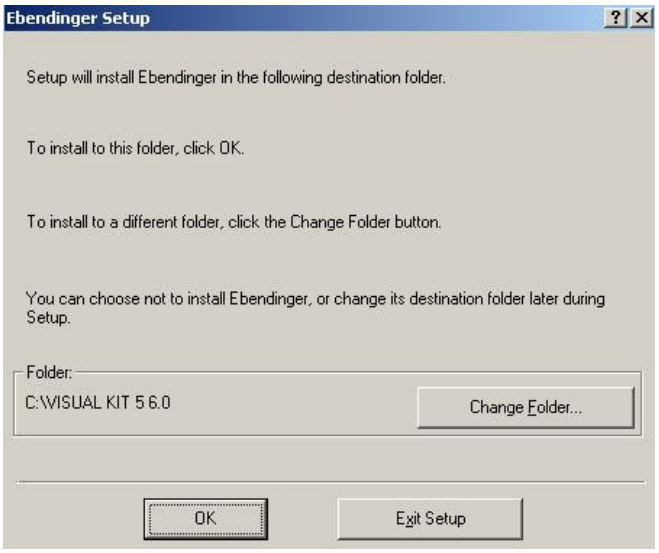


Fig 5 Não altere o diretório do Visual Kit 5. Pressione o botão Ok**, para continuar a instalação.**

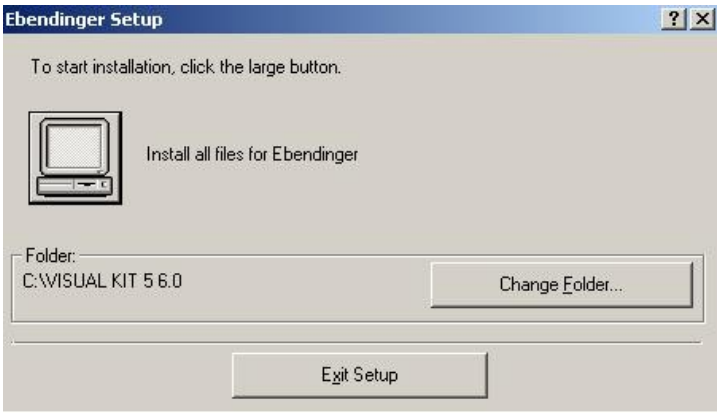


Fig 6 Clique no botão (computador) acima indicado, para continuar a instalação.

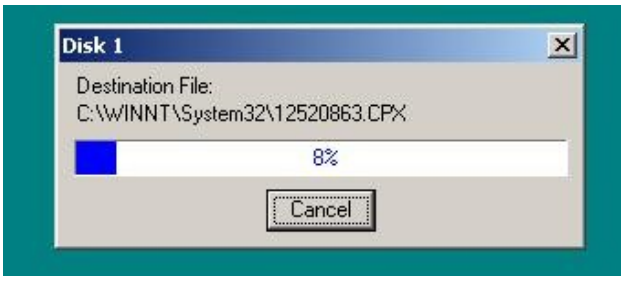


Fig 7 Aguarde enquanto o instalador copia os arquivos para o seu computador.

Fig 8 Pressione o botão Ok**, para concluir a instalação.**

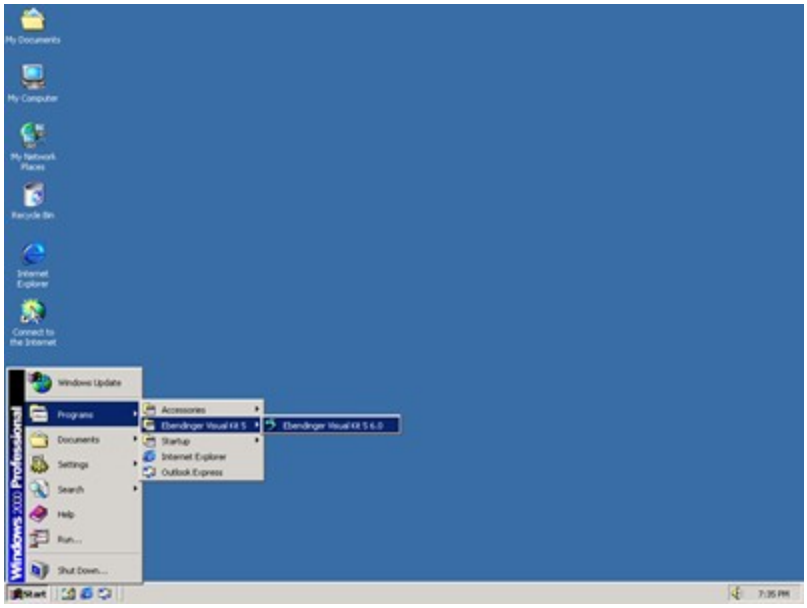


Fig 9 Executar o Visual Kit 5, selecione:

"Iniciar...","Programas","Ebendinger Visual Kit 5", e clique na opção "Ebendinger Visual Kit 5 6.0”

Fig 10 Inserindo número de serie.

Depois de instalado, ao abrir por primeira vez o Visual Kit 5, aparecerá uma janela para inserir o número de serie do produto. Este número de serie se encontra numa etiqueta colada na caixa do CD. Digite com cuidado este número e clique no botão **OK**.

Se aparece a mensagem "**Execute o Visual Kit 5 novamente!**" significa que o número foi aceito. A instalação foi totalmente concluida!. Caso contrario o numero foi digitado de maneira errada, por favor tente novamente.

Razões para não ter conseguido fazer a instalação.

Se você está com dificuldades para instalar o Visual Kit 5, é provável que uma dessas razões seja a causa de seu problema.

Espaço em Disco Insuficiente:

Veja bem se possui espaço livre suficiente para a instalação. Observe que, se estiver instalado em uma unidade comprimida (como as criadas pelo *Stacker*, *DriveSpace* ou *DoubleSpace*), as quantidades estimadas de espaço em disco livre poderão estar sem o desconto da compressão.

Você não consegue acessar o CD:

Normalmente os erros de acesso do CD durante a instalação significam que você precisa encontrar um *driver* de CD atualizado. Veja se está usando a ultima versão de seu *driver* CD.

Seu CD apresenta problemas:

Isso raramente acontece. O sintoma é que o programa de instalação avisa que não consegue encontrar um determinado arquivo. Entre em contato com o serviço ao consumidor da Ebendinger.

O programa de instalação se comporta estranhamente ou não roda:

Normalmente isso é resultado de algum tipo de conflito entre o programa de instalação e outro programa. Feche os programas inicializados automaticamente pelo *Windows*, como protetores de tela, programas de agendas, programas de diretórios *StartUp*, etc. Sai e reinicie o Windows; em seguida tente usar o programa de instalação novamente.

Equipamento defeituoso:

Se o seu equipamento possuir algum problema na memória RAM, *cache*, disco rígido, ou na memória virtual do Windows, a instalação do Visual Kit 5 poderá não se concluir com sucesso.

Você consegue instalar mas não roda.

Em alguns casos, a instalação é concluída com sucesso, mas Visual Kit 5 ou sistemas gerados apresentam problemas na execução.

- Defeitos físicos no computador ou de configuração;
- Está sendo usada uma versão mais antiga do Visual Kit 5 em vez da nova. Leia nesta seção como fazer a atualização;
- Memória insuficiente na sua máquina ou você está rodando muitos aplicativos ao mesmo tempo.

Atenção: Se o problema apresentado se deve a defeitos físicos no seu equipamento contate um técnico de hardware. A Ebendinger não oferece suporte a defeitos de hardware. O suporte é exclusivo ao Visual Kit 5.

Lembre-se:

1) No Windows 98 você não poderá executar as operações OLAP, nem o Data Mining.
2) No Windows 2000 ou superior, você poderá executar o OLAP e o Data Mining, somente se o programa *PTSFULL.EXE* for instalado. O programa *ptsfull.exe* pode ser encontrado no CD-ROM de instalação do Visual Kit 5, diretório \ADO.
3) Para visualizar corretamente todos os elementos que compõe as janelas do Visual Kit 5 e sistemas gerados, configure o seu monitor para **1024 x 768** pontos de resolução.
4) Se você alterar o diretório de instalação do Visual Kit 5, ao tentar executá-lo, a mensagem abaixo será exibida:

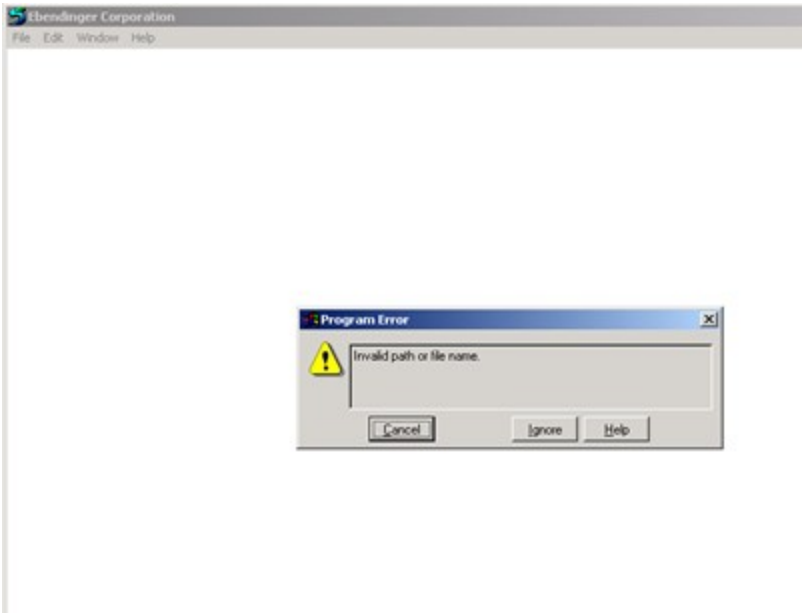


Fig 11 Mensagem, exibida caso modifique o diretório.

Para resolver este problema, execute o ajuste apresentado abaixo:

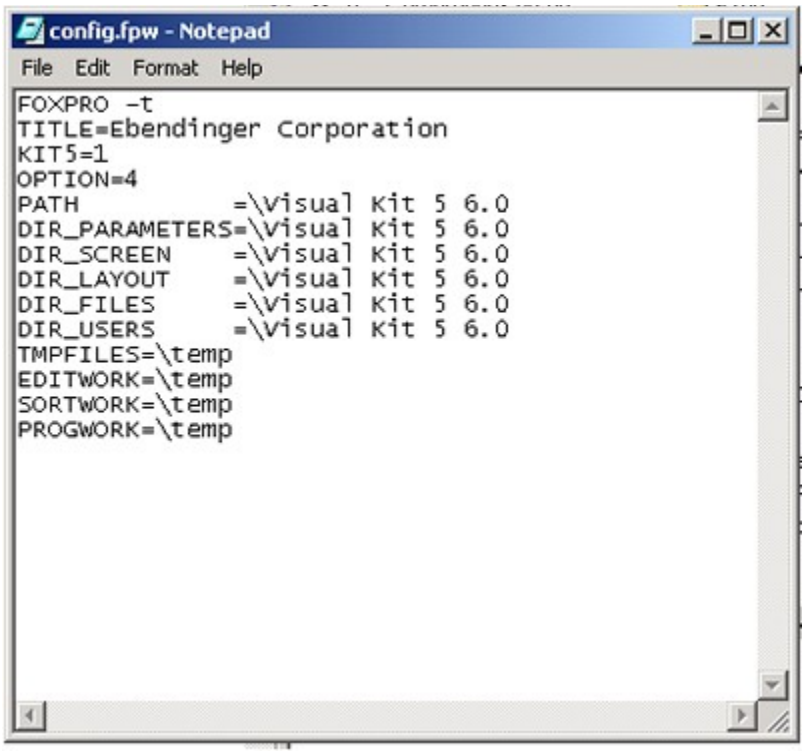


Fig 12 Arquivo Config.fpw aberto pelo *Notepad*.

Abra o arquivo **CONFIG.FPW**, que se encontra no diretório do Visual Kit 5, com o **NOTEPAD** do Windows, e altere os parâmetros como apresentado acima.

Este erro ocorrerá sempre que você copiar um sistema gerado pelo Visual Kit 5, de um diretório para outro, e não ajustar o arquivo **CONFIG.FPW**, informando o novo diretório do sistema gerado.

5) Se o sistema exibir qualquer mensagem, informando que um arquivo **OCX** não foi instalado, significa que você está utilizando uma configuração do *Windows* não apropriada. Veja em Requisitos Técnicos do Visual Kit 5.

II. Conceitos Básicos

Neste capítulo abordaremos várias definições e conceitos que ajudarão a compreender o processo de desenvolvimento de sistemas ...

Sistemas, módulos e entidades.

Há muito, muito tempo atrás, antes do surgimento dos computadores e das bases de dados, os dados eram armazenados em verdadeiros arquivos de aço. Realmente, não faz tanto tempo assim, no máximo duas décadas, mas nesse ínterim assistimos a enormes mudanças tecnológicas na área de hardware e software. Apesar das mudanças, muitos dos conceitos básicos de gerenciamento de dados continuam vigentes. Por isso, para começar, vamos referenciar vários conceitos usados neste manual e no Visual Kit 5 fazendo uso da analogia com aqueles antigos arquivos.

Em primeiro lugar, os arquivos do computador são iguais aos arquivos de aço no sentido da *persistência* - não se perdem quando o programa termina ou quando desligamos o computador... claro, tudo na vida tem um preço - existe o perigo do disco “pifar” ou ser infectado por um vírus, mas da mesma forma que o arquivo de aço está sujeito a perigos iminentes, como fungos e bactérias.

Fig 13 Sistema com dois módulos

O nosso sistema como um todo pode ser representado por esse antigo fichário que aparece na Fig 13.

Cada gaveta deste fichário contém informações de cadastros e contas diferentes e razoavelmente independentes. Claro, está divisão é muito subjetiva e depende muito da visão, da maneira de trabalhar e das regras do dono do fichário. Nos sistemas informatizados acontece algo similar: os sistemas são divididos em sub-sistemas denominados também como “módulos” que ajudam a uma melhor organização e administração, principalmente nos sistemas de grande porte.

Fazendo analogia, no exemplo da Fig 14 teríamos um sistema composto por dois módulos: módulo *Cadastro* e módulo *Contas*.

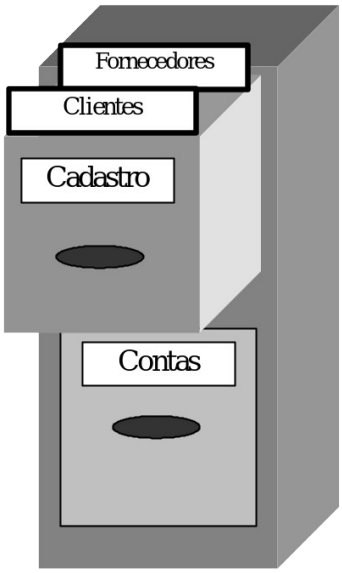


Fig 14Módulo Cadastro com duas entidades

Vejamos agora dentro da primeira gaveta: aí encontramos dois tipos de pastas - a pasta com as fichas dos clientes e a pasta com as fichas dos fornecedores. Estas pastas contêm informações similares mas ao mesmo tempo se diferenciam, cada uma podendo apresentar dados que a outra não possui. Por exemplo, na pasta dos clientes pode aparecer o estado civil e na pasta dos fornecedores o tipo de produto que ele fornece. Junto com as fichas, podem aparecer diversos relatórios associando as informações contidas nas pastas. Exemplo: quantos clientes casados foram cadastrados no último semestre?

Continuando com nossa analogia, estas pastas formam as “entidades” dos sistemas. Temos então no módulo *Cadastro* duas entidades: *Clientes* e *Fornecedores*. Importante ressaltar que os relatórios também podem ser considerados entidades separadas.

Vasculhando o arquivo, abriremos a pasta *Clientes*. Aqui, encontraremos as fichas dos clientes composta por centenas de formulários que contém dezenas de campos preenchidos com informação do tipo: nome completo, sexo, data de nascimento, endereço, etc. Cada formulário descreve um determinado cliente.

Fig 15 Formulários da Entidade Clientes

O formulário é o elemento mais valioso – é ele quem realmente guarda as informações. Tudo é feito para cuidar e salvaguardar os formulários e é por isso, inclusive, que o arquivo é de aço.

No desenvolvimento de sistemas, o mesmo formulário visto acima, é concebido de duas formas diferentes: visual e de armazenamento. No visual, o objetivo é mostrar o formulário na tela do computador de uma maneira amigável para o usuário do sistema e é chamado igualmente de formulário. Do ponto de vista de armazenamento, o formulário vira uma tabela onde cada coluna da tabela representa um campo (entrada de dados) do formulário e cada linha representa um registro preenchido para um determinado cliente.

Visualmente, a entrada e exibição de dados no formulário é feita através de “controles” ou objetos gráficos. Estes objetos são de diferentes tipos e funcionalidades, como: Caixa de Textos (*TextBox*), Lista de Textos

(*ListBox*), Caixa de Seleção (*ComboBox*), Caixa de Edição (*EditBox*), Grupo de Opções (*OptionGroup*), Caixa de Verificação (*CheckBox*); Controle de Rotação (*Spiner*), etc. Mais detalhes sobre os controles são mostrados nos Capítulo V, VI, VII deste manual.

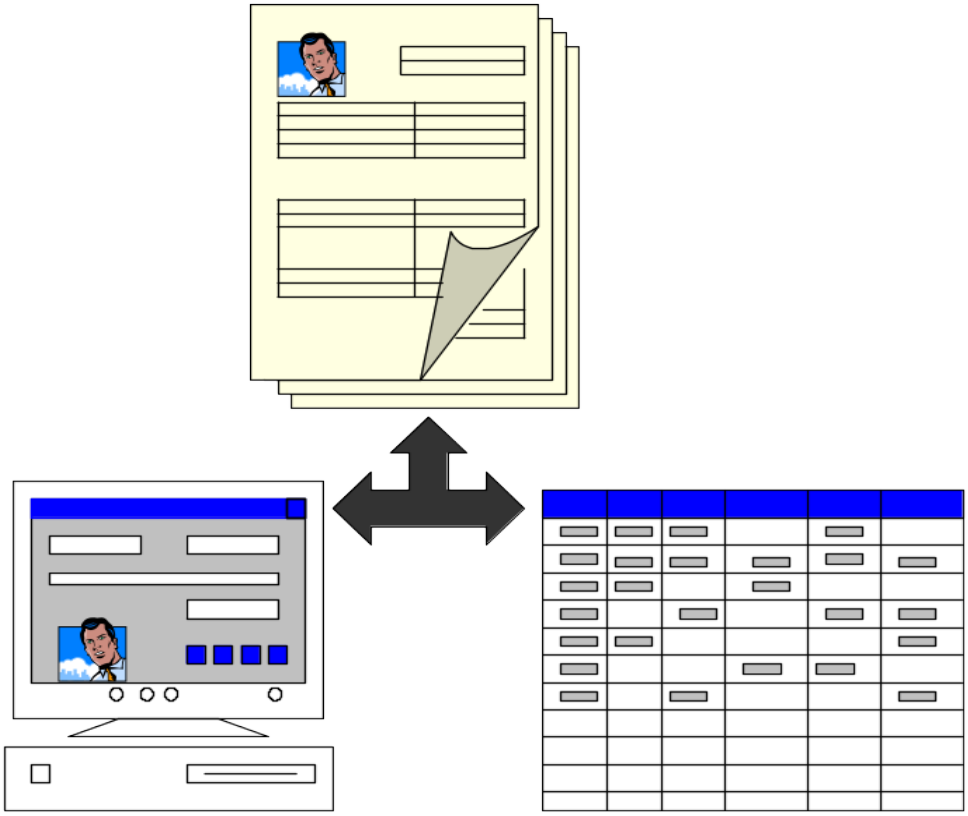


Fig 16 Formulários impressos, formulários e tabelas.

A principal tarefa dos formulários é a distribuição adequada de informações através dos controles ou objetos. No Visual Kit 5 você poderá desenhar os formulários de maneira fácil que automaticamente serão convertidos em tabelas. Cada formulário poderá ser adequado para exibir informações, a fim de permitir a inserção, alteração, exclusão de dados e demais operações e funcionalidades de seu sistema. Estas operações poderão ser realizadas também no formato de tabela.

Na continuação reafirmaremos vários conceitos abordados e colocaremos outros novos mais específicos aos sistemas.

Campos

O campo é a unidade elementar dos dados. Numa tabela, como vimos acima, dividida em colunas e linhas, as colunas seriam o análogo do campo. Geralmente o nome do campo/coluna representa o tipo de informação que ele vai hospedar, ex.: o campo *Data_Nascimento* guardará a data de nascimento do cliente.

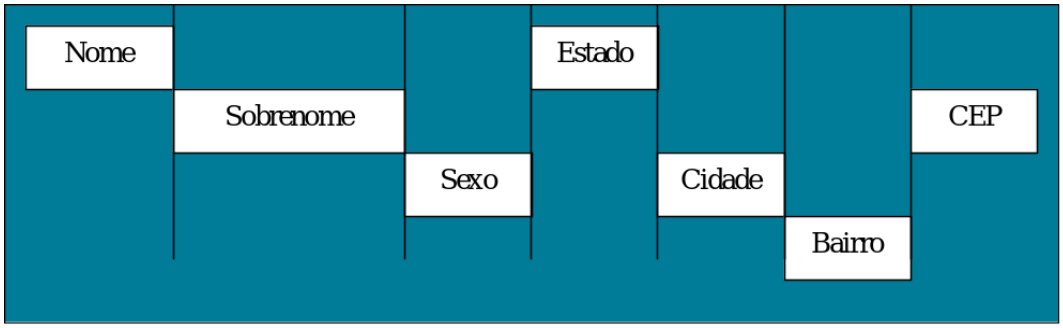


Fig 17 Os campos: unidade elementar de dados

Idealmente os campos devem ser utilizados para guardar peças do dado. Por exemplo a informação nome do usuário pode ser dividida nos campos *PrimeiroNome* e *Sobrenome* em lugar de um único campo *NomeCompleto* ou a informação endereço do usuário pode ser dividida em vários campos: *Estado*, *Cidade*, *Bairro*, *CEP*, *Logradouro*, Número, *Complemento*.

A importância de dividir a informação em unidades elementares de dados pode ser ilustrada neste exemplo: Suponhamos que o desenvolvedor está desenhando um sistema para cadastrar clientes, e decidiu colocar o nome do usuário num único campo chamado *NomeCompleto*. Se depois é requerido de um relatório com a lista dos clientes ordenada pelo sobrenome, não vai ser uma tarefa tão fácil determinar qual parte se refere ao nome e qual parte se refere ao sobrenome. Se o nome do cliente fosse dividido em dois campos *PrimeiroNome* e *Sobrenome* ficaria muito simples criar o relatório ordenando unicamente pelo campo *Sobrenome*.

Registros

Os registros são formados por uma coleção de campos, que representam uma entidade lógica. Numa tabela, as linhas seriam o análogo dos registro, cada linha representa um registro. Por exemplo, o registro com as informações dos clientes contêm os campos *Nome*, *Sobrenome*, *Sexo*, *Estado*, *Cidade*, *Bairro*, *CEP*, *Número, Complemento*. Todos os campos

que compõem os registros são armazenados juntos (de maneira lógica[1]) e representam a informação de um único cliente.

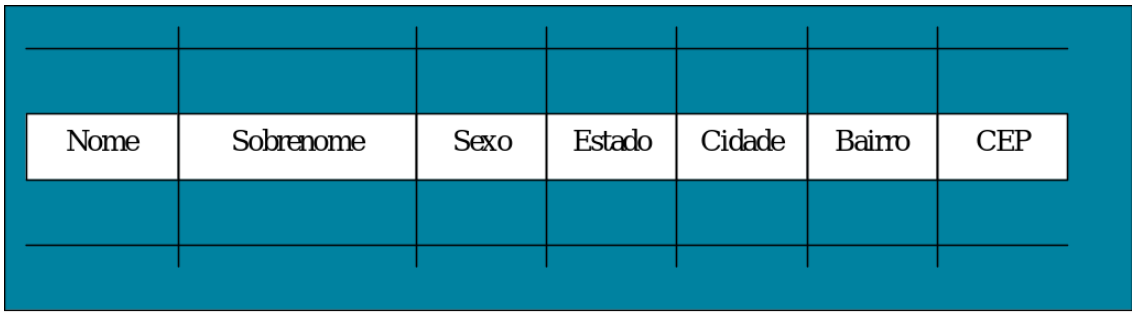


Fig 18 Registro: coleção de campos.

Tabelas

O principal componente de uma base de dados é a tabela. A tabela por sua vez é uma coleção de registros. Da mesma maneira que o nome do campo representa o tipo de dado que ele guarda, o nome da tabela deve representar o tipo de informação que ela contém. Por exemplo a tabela que guarda os endereços dos clientes poderia se chamar *Clientes*.

Na prática, os sistemas guardam mais de uma tabela que representam diferentes informações. O ideal seria que cada tabela tivesse uma única informação. Por exemplo se numa tabela contém o nome do cliente, outra tabela do sistema não precisa ter também essa informação. Existem muitas razões para que isso seja dessa forma, a mais básica de todas é que, se por exemplo o nome do cliente muda, você teria que mudar em várias tabelas. Então, já que todas as tabelas guardam uma única informação, como posso combiná-las para prover uma informação útil? A resposta está no relacionamento entre as tabelas.

Relacionamento

O relacionamento é usado para interligar (*linkar*) duas tabelas e produzir uma informação útil. Duas tabelas são relacionadas entre si através do uso de campos comuns únicos em cada registro. Estes campos são geralmente referenciados como campos chaves.

[1] Fisicamente, no disco rígido do computador esta informação pode estar separada, mas isto é totalmente transparente para os usuários do sistema.

Um desenvolvedor de sistema que não queira usar múltiplas tabelas relacionadas, teria que usar uma única tabela com toda a informação embutida nela. É fácil de perceber que administrar um sistema dessa maneira ficaria muito difícil, além de tornar o sistema muito menos prático em sua utilização. Usando múltiplas tabelas relacionadas para representar dados mais complexos pode parecer complicado, mas sem dúvida, é a melhor maneira de administrar um sistema.

Um-para-Um

Uma das formas de relacionar tabelas é a chamada de relacionamento Um-para-Um (*One-to-One*). Neste tipo de relacionamento um único registro de uma tabela é relacionado exatamente a um único registro de uma outra tabela apenas. Um exemplo de relacionamento Um-para-Um é o seguinte: Suponhamos que temos uma tabela *Clientes* que contem informações do cliente do tipo: nome, endereço, número de telefone, etc. E uma outra tabela que contenha informação relacionada a conta pessoal do cliente com informações de tipo: número da conta, renda, saldo, etc. Cada registro de endereço dos clientes pode ser associado com seu respectivo registro de conta pessoal. Utilizando este tipo de relacionamento podemos descompor grande coleções de informações.

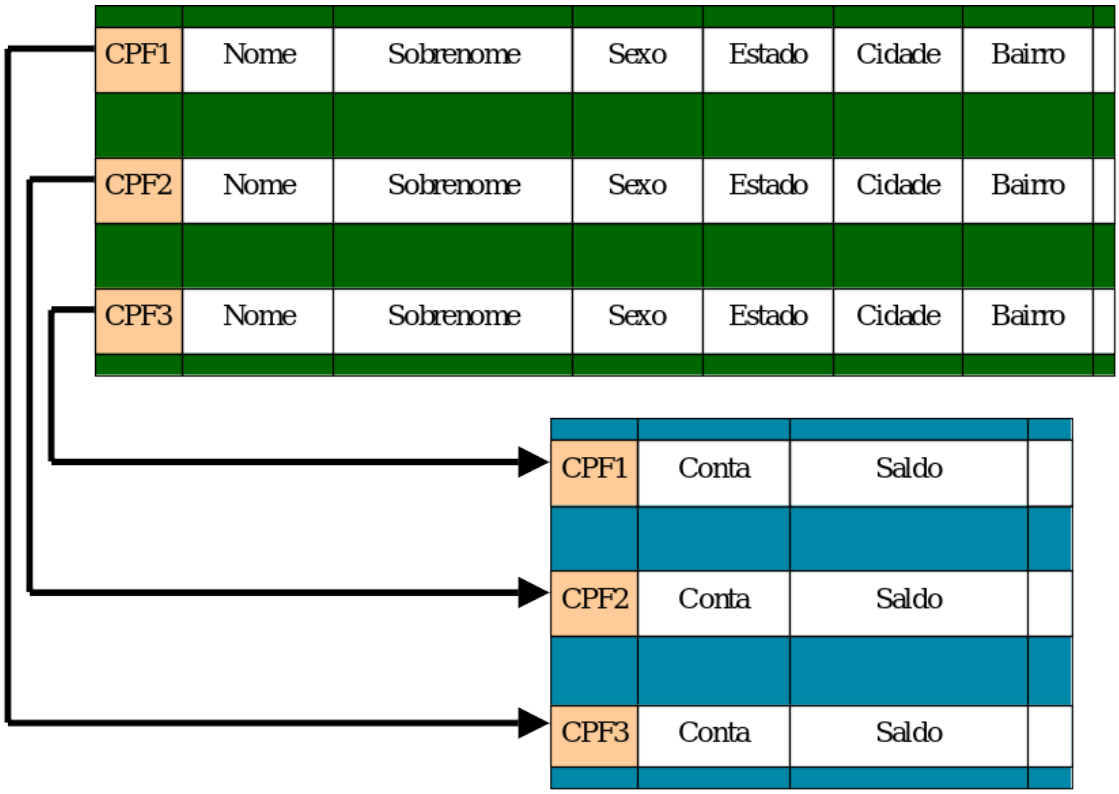


Fig 19 Relacionamento um para um

Um-para-Muitos

Outra forma de relacionar tabelas é chamada de relacionamento Um-para-Muitos (*One-to-Many*). Neste tipo de relacionamento, um registro na tabela é relacionado a múltiplos registros de uma outra tabela. Um exemplo de relacionamento um-para-muitos poderia ser uma tabela principal que contém informação do endereço do cliente e uma tabela secundária que contém informação dos pedidos dos clientes. Neste tipo de relacionamento podem existir mais de um pedido para um mesmo cliente que são relacionadas a um único registro da tabela de endereço de clientes. Desta forma você pode manter e controlar qualquer quantidade de pedidos de clientes sem perder jamais o endereço. Esta flexibilidade é o principal benefício dos relacionamentos.

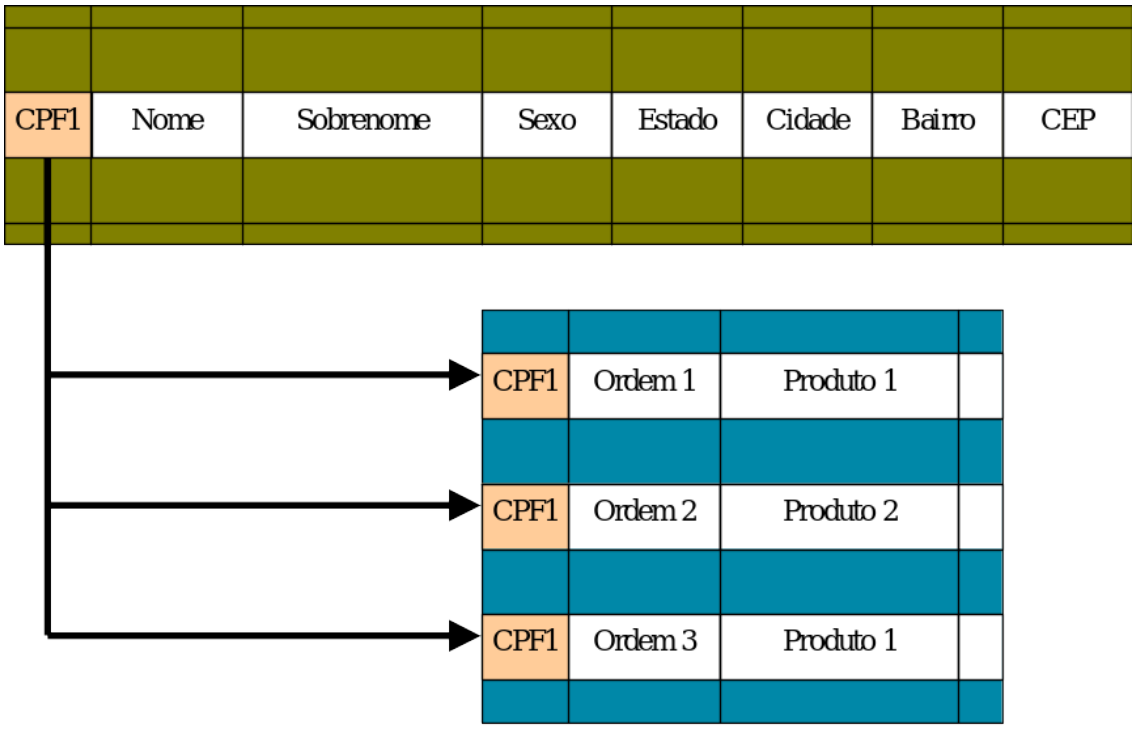


Fig 20 Relacionamento um para muitos

Índices e campos chave

Em um sistema relacional, os índices são usados para ajudar a formar os relacionamentos entre as tabelas e para aumentar a velocidade de acesso aos dados. Um índice é baseado nos campos chave das tabelas. Existem dois tipos de campos chave que uma tabela pode conter: chave primária e chave secundária.

A chave primária é o único campo usado para controlar os registros na tabela. A chave primária deve ser única, ou seja, uma tabela somente pode conter uma chave primária. Em nossa tabela de endereços de clientes a chave primária pode ser o campo *CPF* ou um outro campo que contenha algum código que identifique o cliente, não podendo haver dois clientes com a mesma chave primária. As chaves primárias são compiladas dentro de um outro arquivo chamado de *índice*, o qual acelera o acesso aos dados da tabela. Se você sabe o código do cliente, o índice permitirá a você rapidamente acessar aquele registro na tabela de clientes. Isto é possível porque o arquivo *índice* registra cada campo chave e um ponteiro do registro na tabela.

Campos chaves podem ser usados para formar relacionamentos de uma maneira diferente a explicada acima. Chaves quando usadas simplesmente para juntar duas tabelas são chamadas de “chaves secundárias ou estrangeiras”. Uma tabela pode conter mais de uma chave estrangeira ou simplesmente não ter nenhuma. As chaves estrangeiras são usadas para formar relacionamento entre as tabelas e elas não precisam ser únicas nas tabelas onde radicam.

Operações

É chamado de operações, a tudo que possaa ser feito com os dados armazenados nas tabelas dos sistemas. Existem operações básicas como Inclusão, Alteração e Exclusão de registros sem as quais seria impossível o gerenciamento do sistema. Outras operações tem funcionalidades mais específicas como: Consultas, Geração de Etiquetas, Geração de Relatórios, etc.

O visual Kit 5 brinda uma série de operações que o usuário pode escolher para incluir em cada entidade desenvolvida. Estas operações são:

1- Inclusão
2- Alteração
3- Exclusão
4- Consulta
5- Relatório
6- Etiquetas
7- Indexação
8- Senhas
9- Transferências
10- OLAP
11- Data Mining
12- Consultas avançadas

A explicação das funcionalidades de cada operação do Visual Kit 5 pode ser encontrada no Capítulo XVII

III. O Ambiente de Desenvolvimento

O Visual Kit 5 possui um Ambiente de Desenvolvimento especializado para desenhar sistemas empresariais, pessoais, etc., por meio do qual você poderá criar os módulos , as entidades, as telas dos formulários, relacionamentos entre campos, tabelas, etc. de uma forma rápida e simples sem necessidade de programar, nem ter conhecimentos profundos em informática.

Basicamente o ambiente de desenvolvimento permite realizar as seguintes operações:

- Gerar os arquivos executáveis e complementares do sistema.
- Desenhar a interface amigável do usuário através dos formulários.
- Associar funcionalidades específicas aos controles dos formulários.
- Modificar sistemas da biblioteca de sistemas (sistemas brinde).

Neste capítulo, daremos um passeio pelo Ambiente de Desenvolvimento do Visual Kit 5, apresentando todas as áreas e suas respectivas tarefas. Nos demais capítulos você terá a oportunidade de praticar e utilizar a maioria das possibilidades que o Visual Kit 5 oferece.

Acessando o Visual Kit 5

Para acessar o Ambiente de Desenvolvimento do Visual Kit 5 execute os seguintes procedimentos:

1. Dê um clique no botão iniciar (*Start*) na barra de tarefas (*Task Bar*) do seu Windows (ou pressione <CTRL>+<ESC>).
2. No menu apresentado, selecione Programas (*Programs*).
3. Em seguida, escolha “Visual Kit 5” para abrir a pasta contendo os componentes instalados do produto ou procure o folder intitulado “Visual Kit 5”.
4. Execute o ícone intitulado “PRINCIPAL.EXE’.

Após executar os procedimentos anteriores, será exibido o ambiente, conforme mostra a Fig 21.

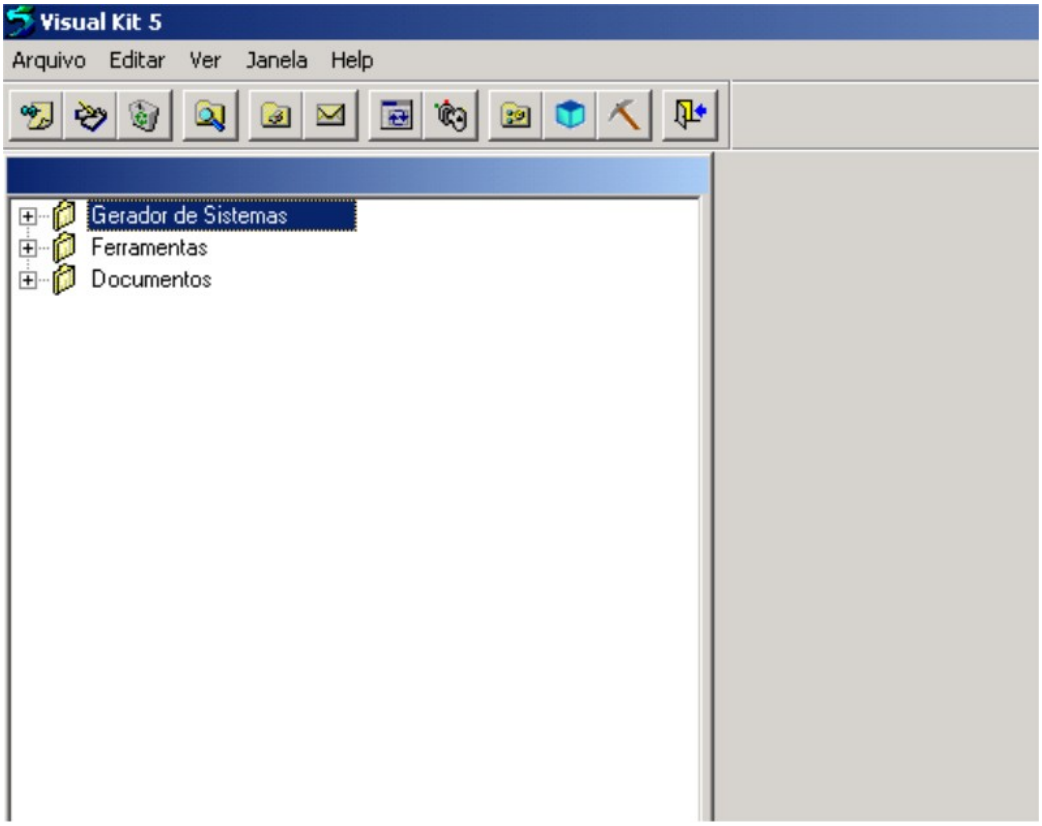


Fig 21 Tela principal do Ambiente de Desenvolvimento do Visual Kit 5.

Elementos do Ambiente

Assim que você inicia uma aplicação Visual Kit 5, é apresentada a janela principal do ambiente que aparece na figura anterior, aparecendo uma barra de menus horizontal, uma barra de ferramentas em formato de botões na parte superior e um menu em formato de árvore (*TreeView*) na lateral esquerda da tela. Veja a seguir uma breve descrição de cada um dos elementos apresentados.

Janela Principal

Esta é a janela superior do ambiente de desenvolvimento do Visual Kit 5. Ela é a responsável pelo gerenciamento de todos os elementos no desenvolvimento. Por exemplo, se você minimizá-la, todos os elementos serão minimizados e fechando-a você estará encerrando a sessão atual do Visual Kit 5.

Barra de Menus

Esta barra apresenta os diversos menus que o Visual Kit 5 possui. Cada menu, seguindo o padrão *Windows*, possui um conjunto de opções distribuídas em subníveis. Cada subnível é caracterizado por uma linha horizontal que separa uns dos outros. As opções de cada menu podem ser visualizadas em duas tonalidades: preto – opções habilitadas, e cinza, opções desabilitadas. As opções desabilitadas indicam que as mesmas não podem ser acessadas antes que alguma tarefa específica seja realizada.

Ainda tratando dos padrões dos menus, opções que possuem reticências indicam que as mesmas não serão executadas imediatamente, isto é, sempre que elas forem acessadas será apresentada uma caixa de diálogo na qual o usuário deverá especificar diversas opções inerentes à execução da tarefa escolhida. Exemplo, se você escolhe a opção *Configurar Impressora* do menu *Arquivo*, a qual permite configurar a impressão. Esta opção apresentará uma caixa de diálogo que lhe permite especificar diversos itens sobre a configuração de impressão, como, por exemplo, o nome da impressora, o tipo de papel, etc.

Dica:

Note que cada menu ou opção possui uma letra sublinhada. Batizadas de mnemônicas. Tais letras, quando pressionadas juntamente com a tecla ALT, permitem acessar a opção em questão sempre que o menu estiver aberto.

Menu Arquivo

Este menu possui cinco opções inerentes a manipulação de arquivos:

- *Abrir*: abre a janela para a criação de um sistema novo, modificação de um já existente ou geração de um arquivo da biblioteca de sistemas (sistema brinde).
- *Fechar*: Fecha sistema aberto.
- *Salvar*: Armazena o sistema atual.
- *Configurar Impressora...*: exibe a caixa de dialogo padrão para configuração de impressoras.
- *Sair*: encerra a seção atual do Visual Kit 5.

Menu Editar

Encontrado em qualquer aplicação *Windows*, este menu contém diversas opções que serão bastantes úteis durante o processo de desenho dos formulários.

Menu Ver

Por meio do menu Ver, você poderá ativar ou desativar diversas opções do ambiente de desenvolvimento

Menu Janela

Este é o menu padrão encontrado na maioria das aplicações *Windows*. Por meio dele você poderá distribuir as diversas janelas apresentadas Visual Kit 5.

- *Organizar Tudo:* Quando se possui mais de uma janela, esta opção faz com que as janelas sejam organizadas da melhor forma possível.
- *Ocultar Tudo:* Oculta as janelas e a *TreeView* (formato de árvore) de todo o sistema.
- *Reexibir:* Exibe as janelas e a *TreeView* de todo o sistema que foram ocultos a opção anterior.
- *Limpar:* Atualiza a tela quando esta se encontra com algum “lixo” na imagem.
- *Em Cascata:* Organiza as janelas abertas do Visual Kit5 em forma de cascata.

Menu Help

Por meio deste menu você pode obter ajuda sobre comandos e procedimentos do Visual Kit 5. Esta ferramenta oferece um conjunto de informações técnicas sobre os diversos produtos da Ebendinger.

Barra de Ferramentas

A barra de ferramentas contém ícones que representam as tarefas mais utilizadas. No decorrer deste manual, farei referência ao uso particular das opções do menu, bem como das ferramentas disponíveis nas barras de ferramentas. Nesta etapa colocarei somente uma breve descrição de cada uma das ferramentas disponível na barra de ferramenta.

Inclusão. Função para registro das informações (entrada).

Alteração. Função para alteração das informações já existentes (incluídas).

Exclusão. Função para excluir os registros (informações) já existentes nas tabelas (entidades) do sistema.

Consultas. Função que permite visualizar o arquivo inteiro, fazer pesquisas, ordenações dos dados, cálculos etc.

Relatórios. Função para impressão ou visualização dos dados de um determinado arquivo.

Etiquetas. Função para impressão de mala-direta.

Indexação. Função para ordenação dos dados de um arquivo. Esta função ordena os dados de arquivos padrão DBF, segundo uma chave pré-definida.

Senhas. Função de segurança. Permite o cadastramento de senhas para cada entidade do sistema.

Transferência. Permite a importação e exportação de dados. Gera ou importa arquivos de diversos padrões, entre eles arquivos Excel.

OLAP (*OnLine Analytical Processing*). Ferramenta para modelo de dados multidimensionais chamados de cubos OLAP.

Data Mining (Mineração ou Exploração de dados). Cria modelos para predição usando algoritmos de inteligência artificial.

Sair. Encerra a execução do sistema.

Menu *TreeView*

Este menu em forma de árvore, descreve os módulos e entidades próprias do Visual Kit 5. O sistema Visual Kit 5 contém três módulos: *Gerador de Sistemas*, *Ferramentas* e *Documentos*.

Módulo de Gerador de Sistemas

Por meio deste módulo o usuário pode acessar a todos os parâmetros do sistema em desenvolvimento.

Sistema Completo

Esta entidade está disponível somente para a operação de “Inclusão”. É nela que são criados e alterados os sistemas completos de maneira rápida e simples.

Módulo

Arquivo onde são armazenados o número e nome dos módulos do sistema a ser gerado ou alterado. Exemplo: 01 - Controle de Estoque.

Entidade

Armazena o número, nome e quais operações estão habilitadas por entidade.
Exemplo:

Módulo: 01
Entidade: 001
Nome: Clientes.

Abertura de Arquivo

Contém as informações sobre quais arquivos serão abertos, as chaves dos arquivos, o servidor de banco de dados, quando for executada uma entidade.

Atributos

Armazena todos os dados necessários para manipulação dos campos de cada arquivo: linha, coluna de exibição do campo na tela, formato, pré e pós-validação, cor, fonte, tamanho, estilo, cabeçalho, mensagem de rodapé etc. Alterar qualquer um destes dados implicará na alteração automática do sistema gerado.

Macro Cálculo

Este arquivo armazena dados da macro cálculo criada no sistema gerado na consulta e relatório.

Macro Relatório

Este arquivo armazena dados da macro relatório, etiqueta e transferência criada no sistema gerado.

Macro Indexação e Filtro

Este arquivo armazena dados da macro indexação e filtro, para consulta, relatório, etiqueta e transferência criada no sistema gerado.

*Chamada Externa Windows *.EXE*

Este arquivo armazena o PATH (caminho de busca) e nome dos programas executáveis que poderão ser chamados sem sair do sistema gerado. Exemplo: C:\WINWORD\WINWORD.EXE.

Pré e Pós Validação

Este arquivo armazena a pré e pós-validação por registro para inclusão, alteração e exclusão de dados. As validações para relatórios, lay-outs, etiquetas e transferências.

Log Alter. / Exclusão do Sistema

Neste arquivo são cadastradas todas as alterações e exclusões realizadas nos dados do sistema gerado. Permite o controle por usuário, data, hora, arquivo e dados alterados ou excluídos. Grava o conteúdo do campo antes e depois da alteração. Grava os registros que foram excluídos além de quem excluiu.

Alias de Referência de Arquivos

Apelidos para referenciar arquivos (tabelas) de base de dados não criada pelo Visual Kit 5.

Cubos OLAP

Armazena os dados para controlar a criação de Cubos OLAP.

Data Mining (DMM)

Armazena os dados para controlar a criação de modelos de Data Mining.

Objetos e Propriedades

Novo arquivo de controle de parâmetros para a montagem das telas no sistema gerado. Quando você desenha uma tela no Visual Kit 5 ele grava neste arquivo os parâmetros para desenhar a tela e validar a entrada de dados.

Modelos de Data Mining

Armazena informações adicionais sobre os modelos de Data Mining.

Módulo Ferramentas

Localizar

Utilize a janela “Localizar” para encontrar dados em todos os arquivos do sistema através de palavras chave sem precisar abrir o arquivo

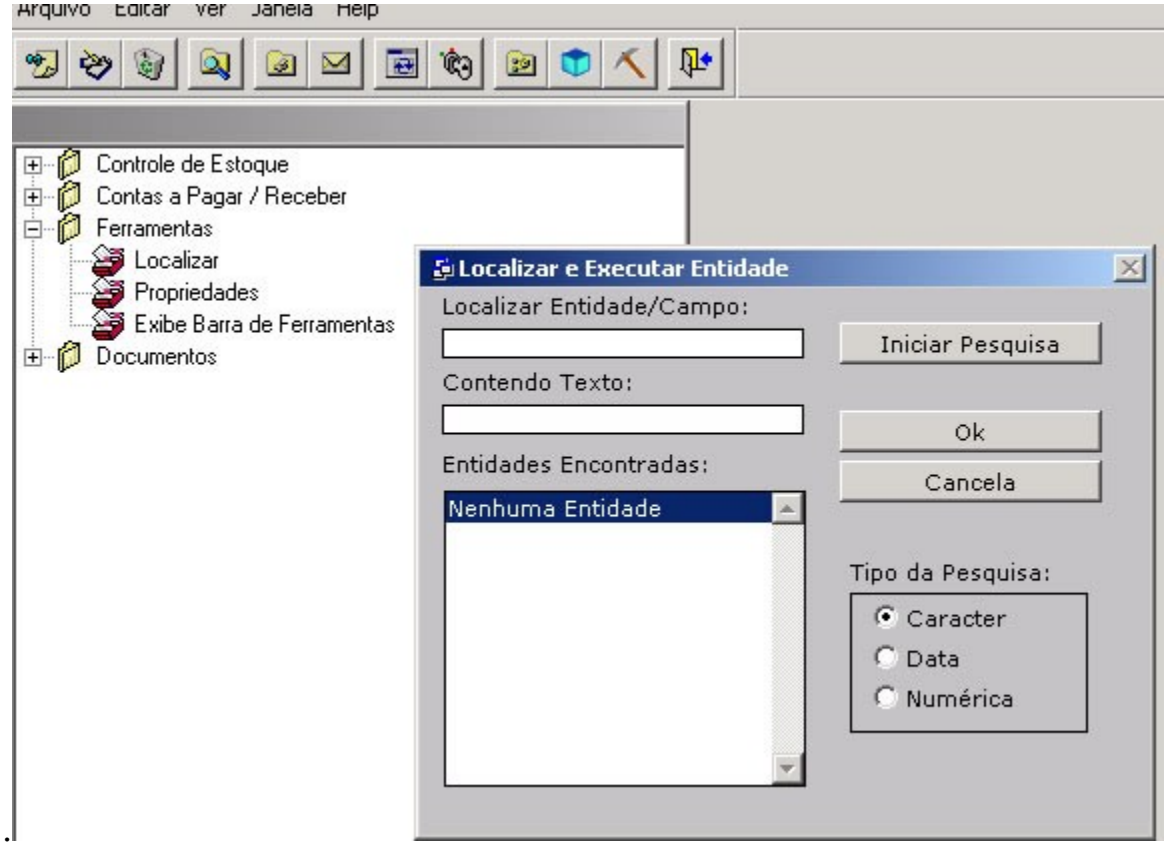


Fig 22 Janela Localizar

Executando o recurso Localizar:

- Feche todos os arquivos abertos;
- Selecione a operação “Consultas”;
- Na *treeview* do sistema gerado, abra "Ferramentas" selecione a opção “Localizar”.
- Localizando uma entidade (arquivo) ou campo de arquivo.
- No campo “Localizar” digite a palavra chave inteira ou um trecho. Exemplo: Clientes ou Cli.
- Pressione o botão “Iniciar Pesquisa”.
- Na lista “Entidades Encontradas” clique sobre a entidade desejada.
- Pressione o botão “Ok”.

Localizando uma informação dentro do arquivo:

- Clique com o mouse no campo “Tipo de Pesquisa”.
- No campo “Contendo Texto” digite a palavra chave inteira ou um trecho. Exemplo: SILVA ou LVA.
- Pressione o botão “Iniciar Pesquisa”.
- Na lista “Entidades Encontradas” clique sobre a entidade desejada.
- Pressione o botão “Ok”.

- Nos dois casos acima descritos, o sistema executará a operação “Consultas” e posicionará o cursor sobre o primeiro registro encontrado.

Propriedades

Utilizando a janela “Propriedades” no sistema gerado, você pode criar usuários para o seu sistema e configurar estes usuários.

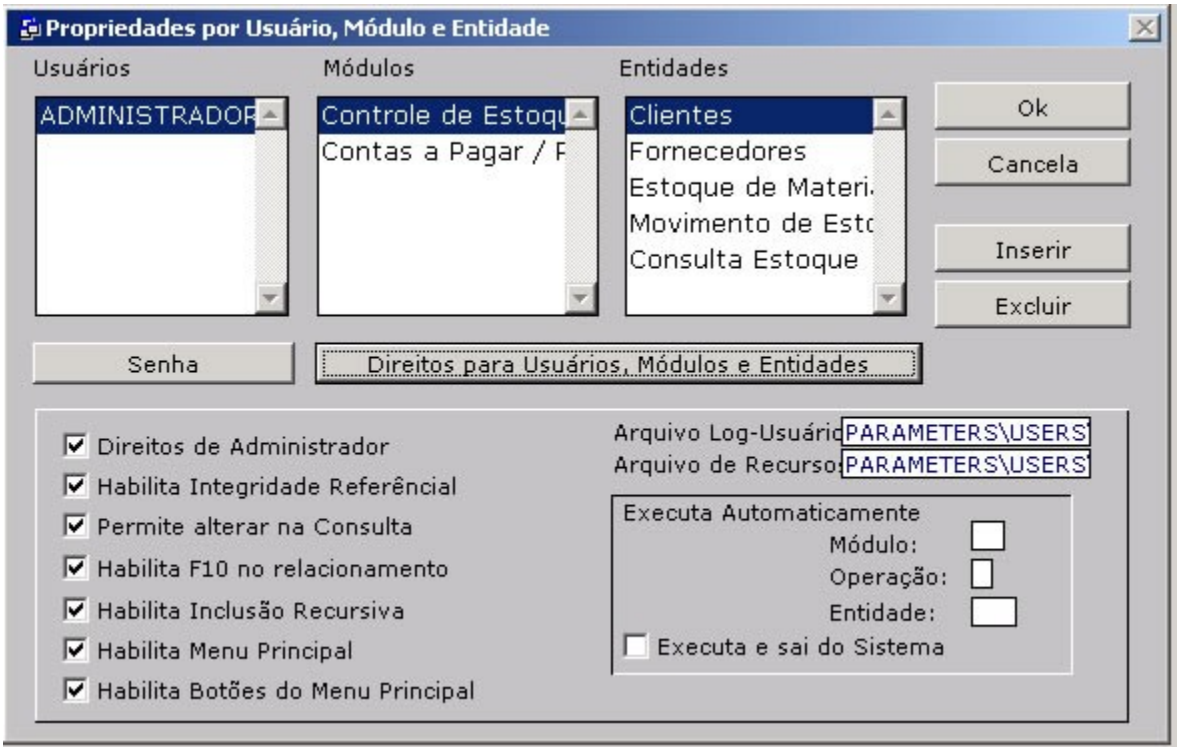


Fig 23 Janela de Propriedades

Atenção !!!

Quando você entra pela primeira vez na janela “Propriedades” o sistema cria de maneira automática um usuário com:

nome (*login*): **administrador**

senha: **admini**

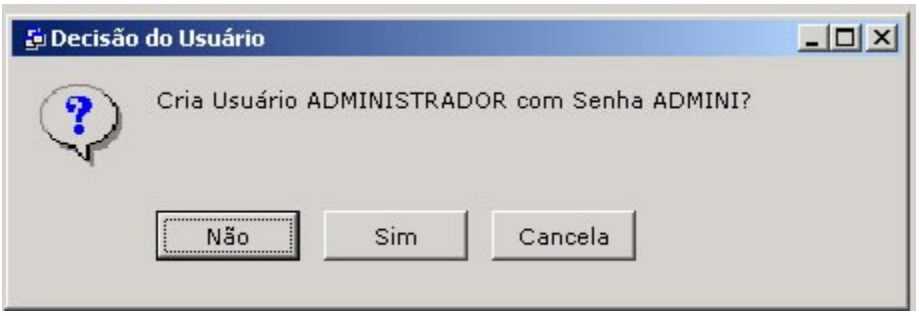


Fig 24 Confirma de decisão de criar senha de Adminstrador.

Executando as Propriedades
Feche todos os arquivos abertos.
Selecione na treeview do sistema gerado o item "Ferramentas" e a opção “Propriedades”.
Criando usuários
Clique na lista “Usuários”.
Pressione o botão “Inserir”.
Digite o nome do usuário e pressione o botão “Ok”.
Se o sistema gerado vai rodar em rede é obrigatório criar todos os usuários do sistema.
Incluindo / alterando a senha do usuário
Selecione o usuário na lista “Usuários”.
Pressione o botão “Senha”.
Digite a senha do usuário e pressione o botão “Ok”.
Excluindo usuários
Clique no nome do usuário na lista “Usuários”.
Pressione o botão “Excluir”.
Confirme a exclusão.
Estabelecendo direitos de usuários, módulos e/ou entidades
Selecione o usuário, módulo e/ou entidade na lista correspondente.
Pressione o botão “Direitos para Usuários, Módulos e Entidades”.
Clique uma vez sobre a operação que você quer desabilitar na janela “Direitos”.
Pressione o botão “Ok” para confirmar os dados.
Desabilitando recursos e menus (configura tela e sistema)
Clique sobre o usuário na lista de “Usuários”.
Clique sobre o recurso/menu que você deseja desabilitar para aquele usuário. A seta indica que o recurso/menu está habilitado.
Criando um arquivo de log por usuário
Clique sobre o usuário na lista de “Usuários”.
Clique no campo “Arquivo Log-Usuário” e digite o path e nome do arquivo (TXT) onde serão gravadas as informações da utilização do sistema.
Exemplo: \WINDOWS\RELLOG.TXT.
Criando um arquivo de recursos

Clique sobre o usuário na lista de “Usuários”.
Clique no campo “Arquivo de Recursos” e digite o path e nome do arquivo (*.DBF) onde serão gravadas as informações sobre os recursos do sistema.
Exemplo: \VDEMOS\FOX01.DBF.
Se a execução do sistema gerado for em rede (multi-usuário) é obrigatório criar um arquivo de recursos para cada usuário do sistema.
Estabelecendo a execução automática de uma entidade
Clique no campo “Módulo” da janela “Executa Automaticamente” e digite o número do módulo.
No campo “Operação” digite o número da operação “1 - para Inclusão”, “2 - para Alteração” etc.
No campo “Entidade” digite número da entidade que será executada automaticamente.
Para sair após a execução do sistema, clique sobre o campo “Executa e sai do sistema”.
Sai da janela “Propriedades” clicando sempre no botão “Ok”.

Exibe Barra de Ferramentas

Mostra ou oculta a barra de ferramentas do sistema.

Módulo Documentos

Neste módulo serão apresentados os últimos 10 (dez) arquivos recentemente manipulados. Isto será muito útil quando você tiver que acessar rapidamente operações e arquivos que foram abertos recentemente.

Acabamos de conhecer as áreas básicas do ambiente de desenvolvimento do Visual Kit 5. Muito provavelmente nos próximos capítulos você terá que retornar a este para lembrar ou localizar uma determinada opção.

A manipulação dessas opções será apresentada de forma particular e com exemplos nos próximos capítulos. Na maioria das vezes, será utilizado apenas um conjunto básico de opções, além de serem bastante fáceis de se utilizar. Para demonstrar que você não terá dificuldades na utilização das opções do Visual Kit 5 vamos agora desenhar o nosso primeiro sistema.

IV. Criando seu Primeiro Sistema Passo a Passo

Até o momento, você já passou por três capítulos e nada de desenvolvimento. Já está na hora de ver como isto funciona!

Obviamente, você ainda não possui toda a prática do mundo nem conhece todos as funcionalidades que o Visual Kit 5 brinda, mas para que você tenha o "gostinho" de desenvolver seu primeiro sistema, vamos juntos desenvolver um sistema bem simples que o ajudará a entender melhor os conceitos e elementos apresentados nos capítulos anteriores, bem como nos próximos capítulos deste manual.

Gostaríamos de colocar que alguns recursos executados neste capítulo não serão abordados em profundidade, pois o único intuito aqui é o de oferecer uma leitura clara dos próximos capítulos.

A proposta

Para desenvolver um sistema, o primeiro passo é entender o contexto do mesmo. Desta forma, poderemos criar mecanismos mais simples tornando o sistema mais claro para o usuário final. Esta é uma regra fundamental que todo desenvolvedor deve cumprir; e agora você é um desenvolvedor. Claro que neste exemplo a coisa será muito mais simples; no entanto, não podemos pular esta etapa de desenvolvimento.

O sistema que desenvolveremos é de controle dos aniversariantes. Trata-se de uma pequena interface contendo apenas um formulário com dois campos. Nesta interface básica, o usuário colocará o nome da pessoa, a data de nascimento.

Desenhando o sistema

O Visual Kit 5 oferece diversos recursos para desenhar sistemas. Como nosso objetivo neste capítulo é apenas mostrar o quanto você é capaz sem profundos conhecimentos do Visual Kit 5, utilizaremos aquele tradicional método de receita de bolo.

Carregando o Visual Kit 5

Em primeiro lugar, você deverá abrir o Visual Kit 5, conforme explicado no capítulo anterior.

Gerando o diretório do sistema

Pois bem, agora que você já está dentro do Visual Kit 5, criaremos primeiramente o diretório o qual guardará todos os arquivos do nosso primeiro sistema. Execute o procedimento a seguir:

- Clique na *TreeView* para abrir a pasta “Gerador de Sistema”, como na figura abaixo.
- Clique em “Sistema Completo” e selecione a opção “Inclusão”. Estaremos incluindo um sistema. Veja na Fig 25.

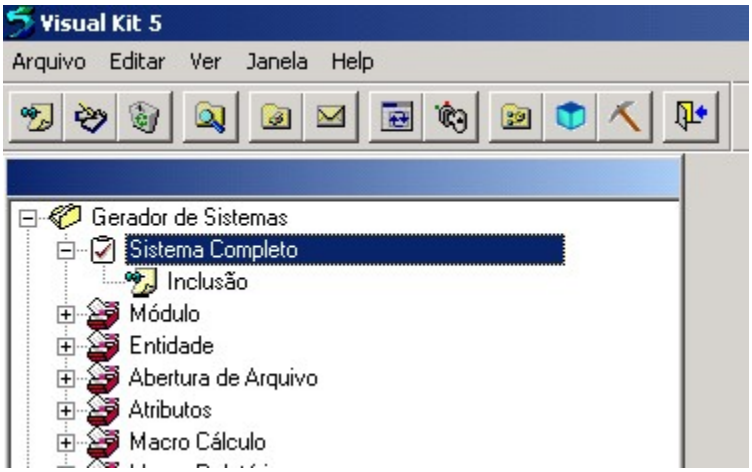


Fig 25 Incluindo sistema

O Visual Kit 5 permite desenvolver sistemas começando de zero ou a partir de uma biblioteca de sistemas prontos (sistemas brinde). Por isso é apresentada uma caixa de diálogos intitulada “Utiliza Sistemas Prontos / Cria Sistema Novo” conforme mostra a Fig 26.

- Como vamos desenvolver um sistema desde o zero, escolhemos a opção "Cria Sistema Novo".
- Selecione seguidamente a opção "Sistema Completo".

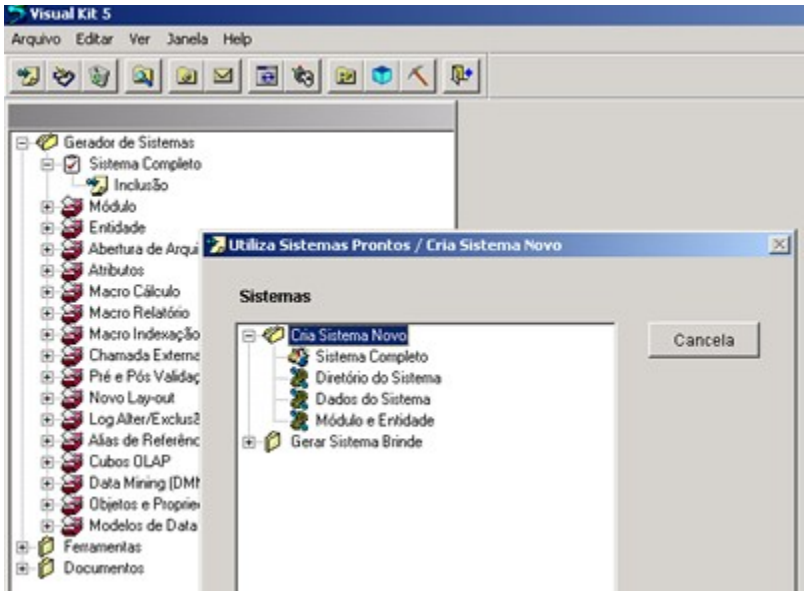


Fig 26 Cria sistema novo.

Ao clicar na opção "Sistema Completo" o Visual Kit 5, abrirá uma janela, solicitando que você digite o nome para o diretório onde será gerado o sistema novo. Digite **\VCLIENTE**

Atenção! Você não pode gerar sistema no próprio diretório do Visual Kit 5 nem como sub-diretório do Visual Kit 5.
Exemplo de diretório *correto*: **\VCLIENTE**
Exemplo de diretório *inválido*: **\VISUAL KIT 5 6.0**
Exemplo de diretório *inválido*: **\VISUAL KIT 5 6.0\VCLIENTE**

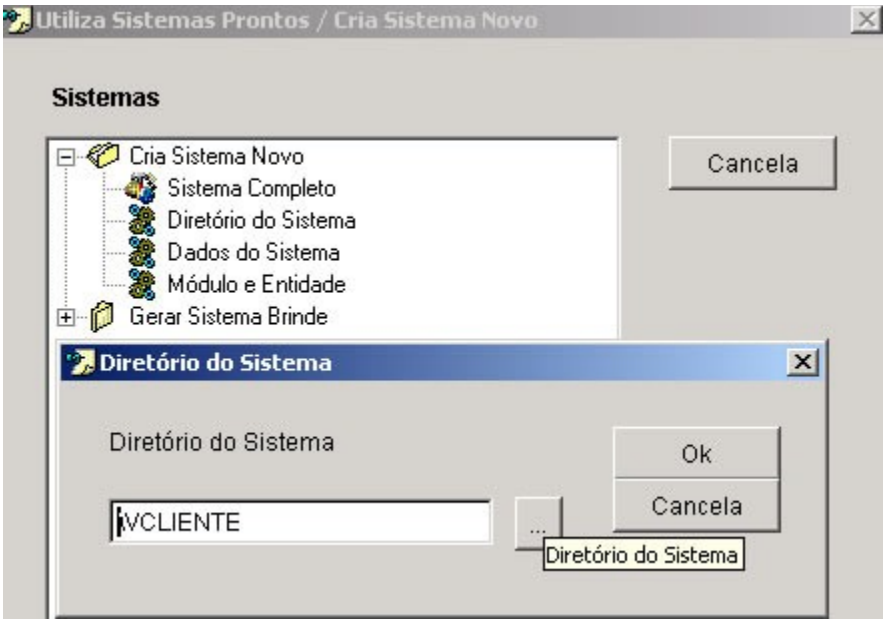


Fig 27 Definindo diretório onde será gerado o sistema.

O Visual Kit 5 exige a confirmação para poder criar o diretório selecionado. Responda que "Sim" / "Yes", para criar o diretório do sistema novo. E pronto, o diretório está criado!.

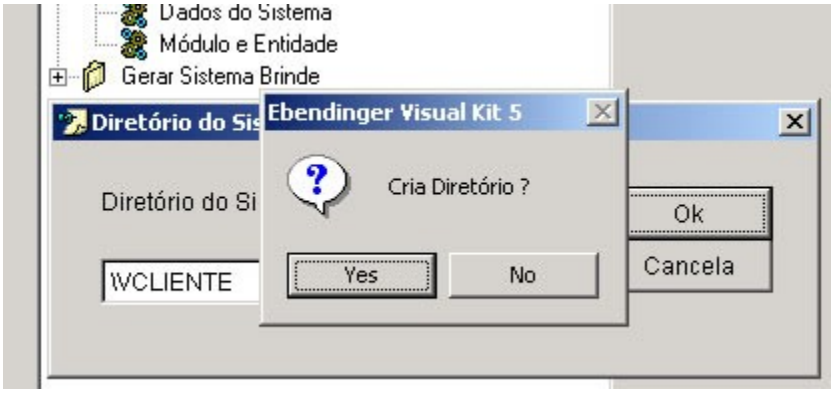


Fig 28 Confirmando criação do diretório do sistema.

Seguidamente aparecerá uma janela para o cadastramento dos dados gerais do sistema, veja na Fig 29. Coloque o “Nome do Desenvolvedor” (você !) , “Nome do Sistema”, “Nome do Cliente”, “Rodapé dos Relatórios”, “Versão do Sistema”, “Idioma”, “Site” e “Driver”. O preenchimento nesta etapa é obrigatório, mas os valores poderão ser mudado posteriormente.

Mantenha na opção "Driver" em **Visual FoxPro**. Não troque por enquanto esta opção!. Vai indicar o tipo de formato das tabelas geradas.

Nota.

O Visual Kit 5 gera os sistemas baseados em bases de dados em dois formatos: Visual FoxPro ou MS SQL Server.

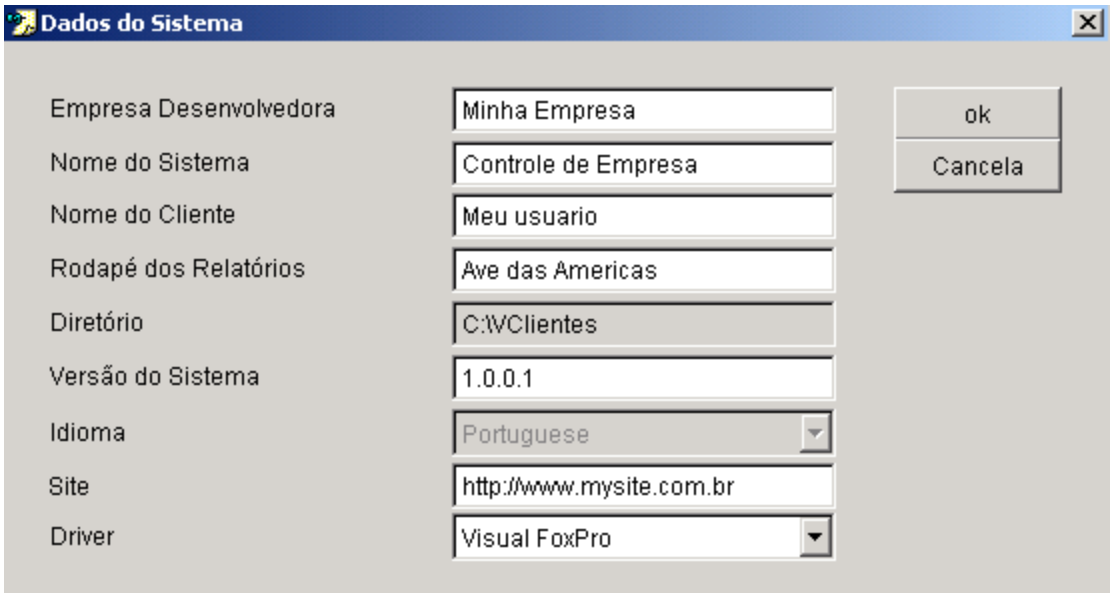


Fig 29 Dados Gerais do Sistema.

Criação dos módulos e entidades

Bom, até agora preparamos as condições iniciais para criar nosso sistema, chegou a hora da própria criação dos módulos e suas entidades. Se tudo correu bem, devemos ter agora na tela do computador a janela de criação dos Módulos e suas Entidades, como mostra a Fig 30

Segundo já foi explicado no Capítulo II, a estrutura dos sistemas criados pelo Visual Kit 5 é constituída por módulos e entidades. Neste exemplo criaremos a variante mais simples de um sistema: um único módulo contendo uma única entidade

Para criar o módulo basta:

- Clicar no botão “Cria Módulo”;

- Aparecerá uma entrada de dados na parte inferior da janela **Módulos** onde digitaremos o nome do módulo “Cadastro”;
- Tecle <ENTER> ou botão “OK”. Será aberta uma janela de confirmação, selecione “Sim” – “Yes”, para confirmar o nome do módulo. Veja na Fig 31.

Nota.

No Visual Kit 5 os módulos são enumerados na ordem de criação, ex. 01, 02, 03... etc. A quantidade máxima de módulos suportada pelo Visual Kit 5 é 99.

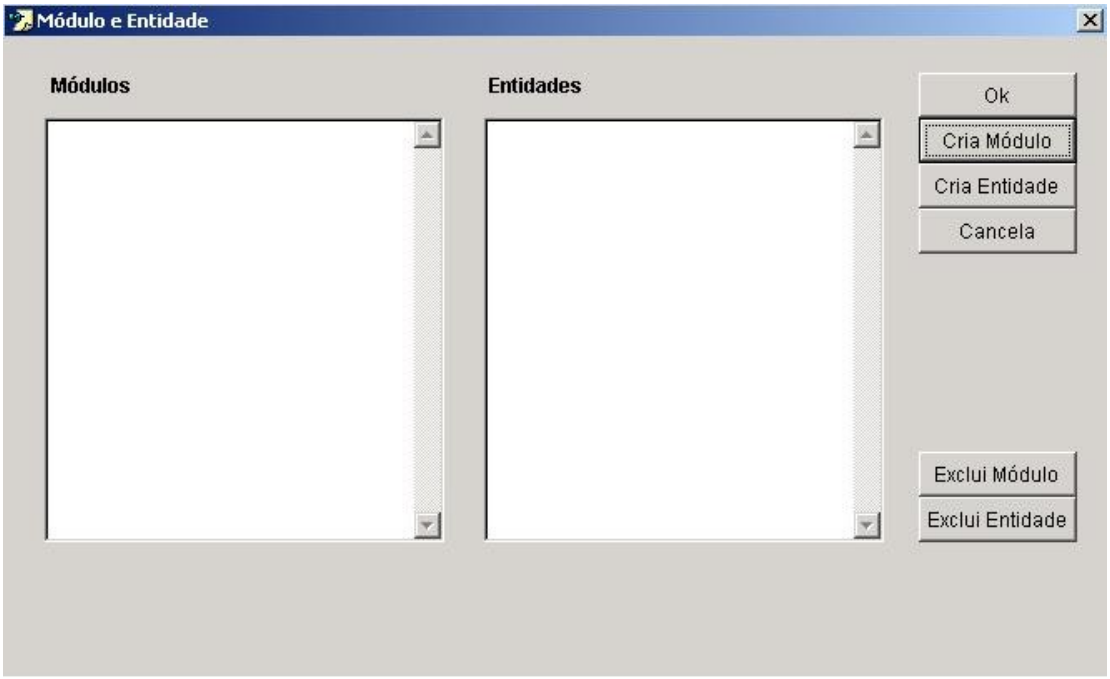


Fig 30 Janela de Definição das estruturas Módulos e Entidades do sistema.

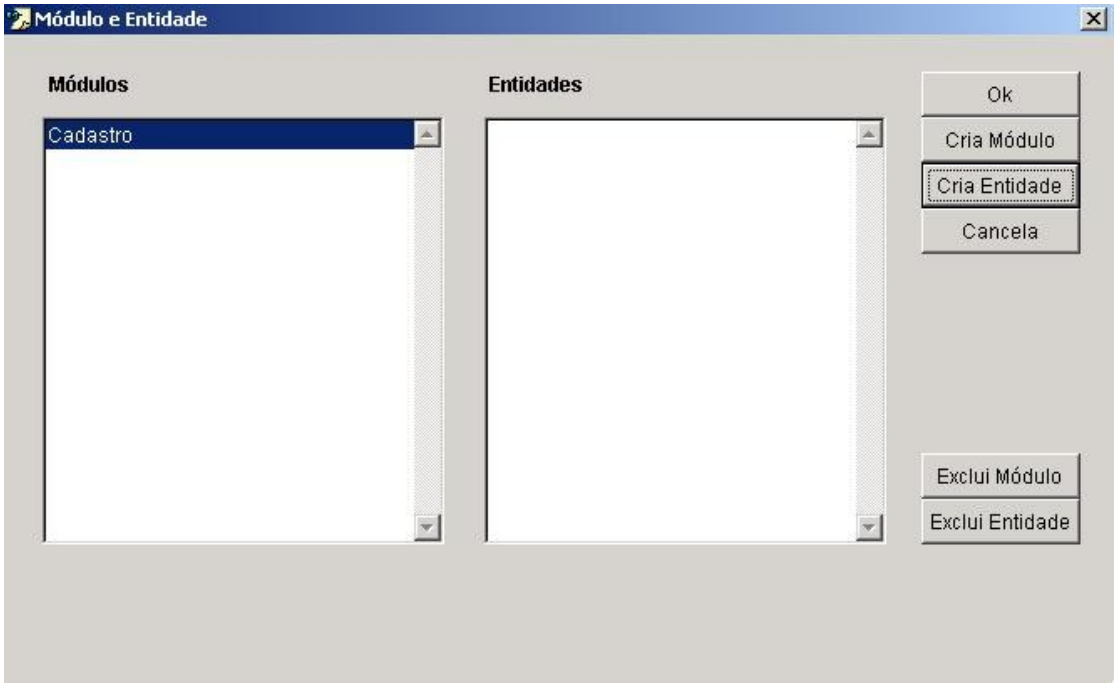


Fig 31 Definindo o primeiro módulo (01): *Cadastro*.

Em seguida cadastre a entidade do módulo. Para cadastrar as entidades de um determinado módulo, clique sempre no nome do módulo e em seguida pressione o botão "Cria Entidade". Aparecerá uma entrada de dados na parte inferior da janela **Entidades** onde digitaremos o nome da Entidade “Clientes”;

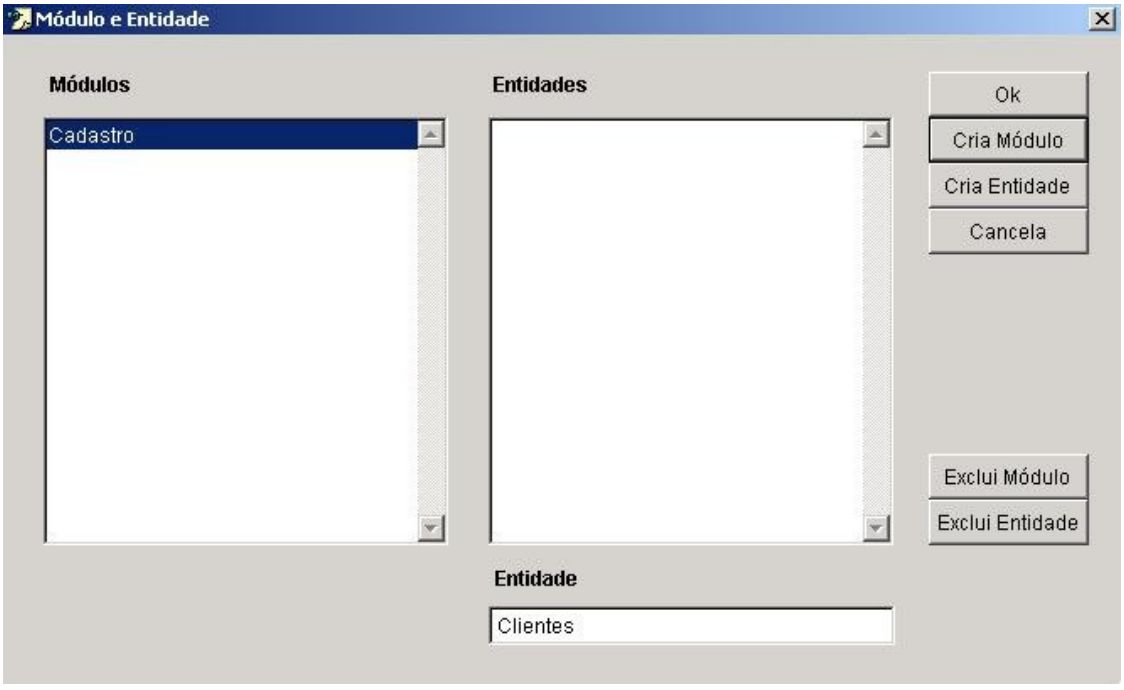


Fig 32 Criando a Entidade

Atenção! Antes de criar uma entidade, selecione sempre módulo clicando sobre o seu nome na Lista de Módulos.

Sempre que você definir uma nova entidade o Visual Kit 5 solicitará a definição das operações básicas que serão herdadas por meio da janela mostrada na Fig 33. Vamos confirmar neste todas as operações básicas do Visual Kit 5 pressionando o botão "Ok".

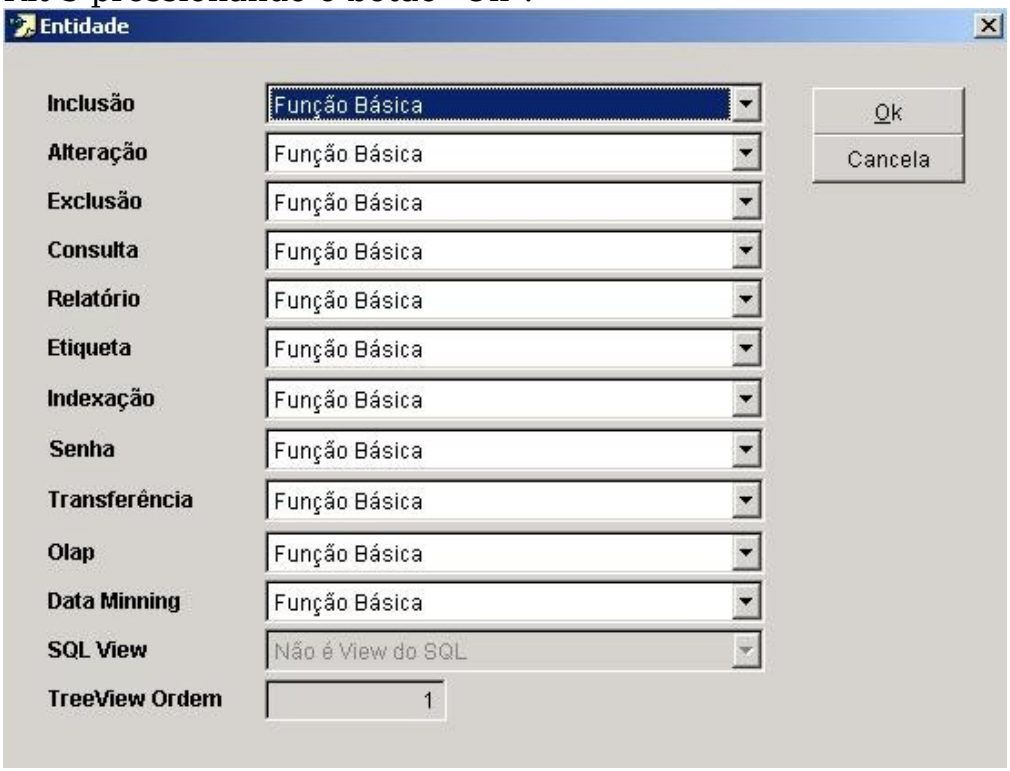


Fig 33 Definindo Operações Básicas que a entidade vai possuir.

Nota.

No Visual Kit 5 as entidades são enumeradas na ordem de criação, colocando primeiro o número do módulo a que elas pertencem. ex. **01**001, **01**002, **02**001...a quantidade máxima de entidades é 999 em cada módulo .

Chegou a hora de desenhar o formulário associado à entidade, sem dúvida a parte mais criativa. Para criar ou alterar o formulário de uma entidade, sempre faça da seguinte maneira:

- Clique primeiramente no nome do módulo que contém a entidade;
- Clique depois no nome da entidade para selecioná-la;

- Clique no botão **Ok;**
- E confirme a seleção.

Desenhando o formulário.

Depois que foi escolhida a entidade *Clientes*, aparecerá três elementos: uma tela de formulário que receberá posteriormente os controles ou objetos e mais duas janelas: a **Caixa de Ferramentas** e a janela de **Propriedades**. Veja na Fig 34.

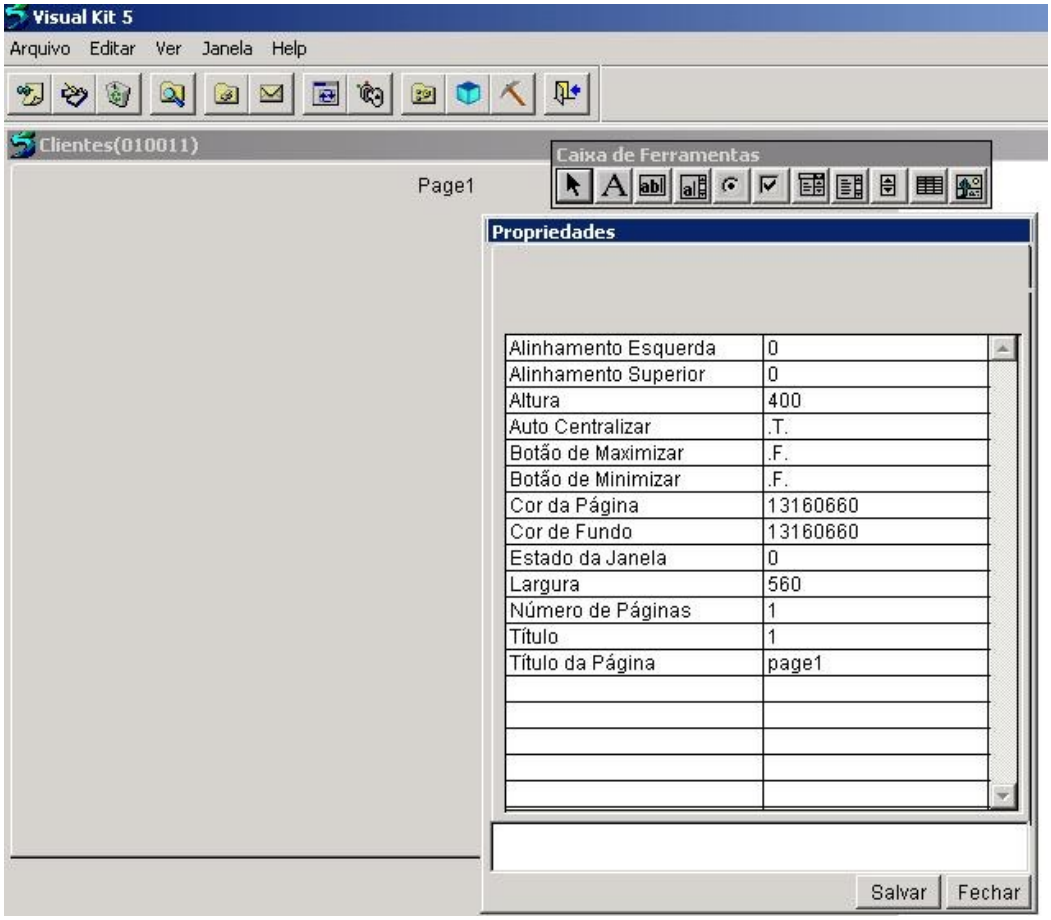


Fig 34 A tela do formulário básico e os recursos de desenho.

Por meio da caixa de ferramentas podemos escolher e inserir os controles nos formulários, para gerar a interface amigável para o usuário de nosso sistema.

Já as propriedades são as características relacionadas a cada controle ou objeto. O principal papel das propriedades é estabelecer como os controles devem ser apresentados e ou manipulados na interface da aplicação.

Inserindo os controles.

Todos os objetos dentro do Visual Kit 5 possuem nomes. Por *default*, cada controle possui um nome que identifica categoria da qual o mesmo faz parte. Neste capítulo utilizaremos o nome *default*, mas em capítulos posteriores você aprenderá que devemos alterar tais nomes para que o desenvolvimento dos sistemas fique mais claro.

Neste nosso primeiro sistema, utilizaremos 2 controles básicos *Label* e *TextBox*. Veja a seguir os procedimentos necessários para inseri-los e movê-los dentro do formulário:

Dica.
Para identificar um determinado controle dentro da caixa de ferramentas, mantenha o ponteiro do mouse sobre o controle desejado por alguns segundos. Será apresentada uma caixa de texto amarelada (conhecida como *ToolTip*), contendo o nome da classe do controle em questão.

- Dê um clique no controle *Label* (Rótulo ou texto fixo) na caixa de ferramentas. Note que o botão será diferenciado do resto dos botões sinalando que ele está ativado.
- Seguidamente usando o ponteiro do mouse escolhemos aproximadamente o lugar dentro do formulário onde ficara o *Label*.
- Clique com o botão do mouse e o controle aparecerá com as marcas de seleção (pequenos quadrados negros).

Agora podemos alterar suas propriedades “**Alinhamento Esquerda**”, “**Alinhamento Superior**” para posicioná-lo no lugar exato dentro do formulário e “**Descrição do Título**” para definir o texto que o mesmo apresentará. Veja os procedimentos necessários.

- Na janela de propriedades localize e clique na propriedade “**Alinhamento Esquerda**”. Aparecerá uma entrada de dados na parte superior da janela.
- Digite o valor **60** no conteúdo da entrada de dados, assim o valor será assumido pela propriedade “**Alinhamento Esquerda**” e pressione ENTER.

- Em seguida, localize a propriedade "**Alinhamento Superior**" na janela de propriedades e clique sobre ela. Aparecerá novamente a entrada de dados na parte superior.
- Digite o valor **30** como conteúdo da propriedade "**Alinhamento Superior**" e pressione ENTER.
- Localize agora a propriedade "**Descrição do Título**" na janela de propriedades.
- Substituía o valor *default* digitando *Nome do Cliente***:** como conteúdo desta propriedade.

A Fig 35 exibe o formulário após a execução dos procedimentos anteriormente mencionados.

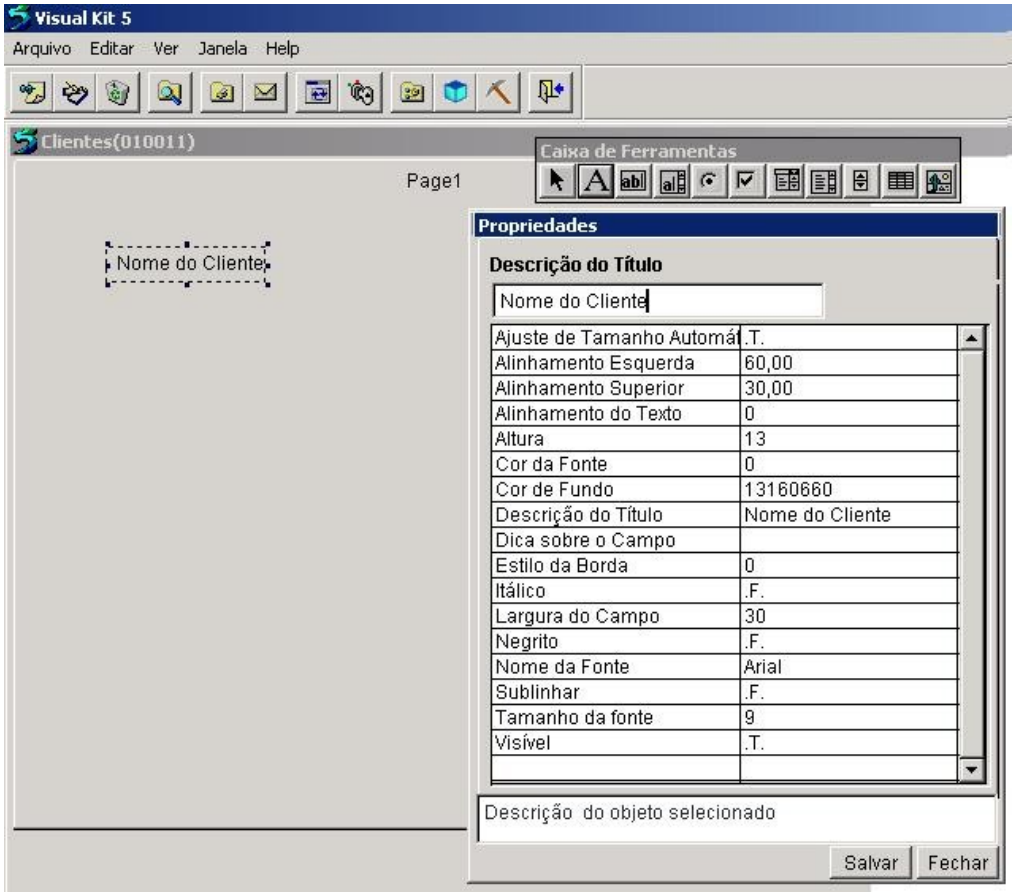


Fig 35 Inserindo o primeiro controle

Muito bom! Agora já temos um rótulo que informa ao usuário do sistema que ele deverá digitar o nome do cliente. Agora falta colocar a própria entrada de dados para o nome do cliente. Utilizaremos o controle tipo *TextBox* ou Caixa de Texto. Veja os procedimentos necessários.

- De um clique no controle *TextBox* na janela de controles. Note que ele ficou ressaltado na janela de ferramentas.

- Usando o ponteiro do mouse escolhemos aproximadamente o lugar dentro do formulário onde ficará o *TextBox*.
- Dê um clique com o botão do mouse e o *TextBox* aparecerá na tela do formulário.

Nota

Note que na janela de propriedades existem propriedades com um fundo azulado e propriedades com fundo branco. As propriedades com fundo azulado são as relacionadas à estrutura dos dados e as propriedades de fundo branco são as relacionada estritamente com a estética ou parte visual do formulário.

- Localize a propriedade "**Alinhamento Esquerda**".
- Digite o valor **180** como conteúdo da propriedade "**Alinhamento Esquerda**" e pressione ENTER.
- Localize a propriedade "**Alinhamento Superior**". Digite o valor **30** e pressione ENTER.

Para alterar as dimensões de um objeto, basta de arrastar suas bordas no sentido desejado. Embora seja esta a forma mais fácil, utilizaremos outra.

- Utilizando a barra de rolagem vertical da janela de propriedades, localize a propriedade "**Largura do Campo**" e dê um clique nela. Esta propriedade especifica o número máximo de caractere que poderá ser digitado .
- Digite o valor **220** como conteúdo e pressione ENTER. Observe como aumentou a largura do campo.

Da maneira similar, incluiremos mais um controle *Label* e um controle *TextBox* para colocar a data de nascimento do cliente. Veja os procedimentos necessários.

- Dê um clique no controle *Label*.
- Usando o ponteiro do mouse escolha aproximadamente o lugar dentro do formulário onde ficará o Label (logo abaixo do *Label* anterior).
- Clique na propriedade "**Alinhamento Esquerda**".
- Digite o valor **60** como conteúdo da propriedade "**Alinhamento Esquerda**" e pressione ENTER.
- Na propriedade "**Alinhamento Superior**" digite o valor **80** como conteúdo e pressione ENTER.
- Localize agora a propriedade "**Descrição do Título**" na janela de propriedades.

- Digite **Data de Nascimento:** como conteúdo desta propriedade.
- Dê um clique no controle *TextBox* na caixa de ferramentas.
- Usando o ponteiro do mouse escolhemos aproximadamente o lugar dentro do formulário onde ficará o *TextBox* (do lado do *Label*).
- Dê um clique com o botão do mouse e o *TextBox* aparecerá na tela do formulário.
- Localize a propriedade "**Alinhamento Esquerda.** Digite o valor **180** como conteúdo da propriedade e pressione ENTER.
- Localize a propriedade "**Alinhamento Superior**". Digite o valor **80** e pressione ENTER.

Até aqui tudo foi muito similar. Agora modificaremos uma nova e importante propriedade: "**Tipo do Campo**" que coloca o tipo de dado que o campo armazenará. Localize esta propriedade e dê um clique nela. Observe que na parte superior aparecerá uma entrada de dados com a letra **C** como *default*. Clique nela e será apresentado para seleção 6 (seis) tipos de campos. Em nosso exemplo escolheremos o tipo **D- Data**:

Nota

Os tipos de campos no Visual Kit 5 são indicados da seguinte forma:

C- Caráter: Qualquer texto.
N- Numérico: Números inteiros ou frações.
L- Lógico: Valor booleano de Verdadeiro ou Falso.
M-Memo: Campo de tamanho variado.
G- Geral: (foto ou objeto OLE):
D- Data: data cronológica constituída de dia, mês e ano.

Por enquanto você terminou a fase de desenho deste primeiro sistema, e precisa agora salvá-lo, você ficaria muito triste se a força caísse agora.

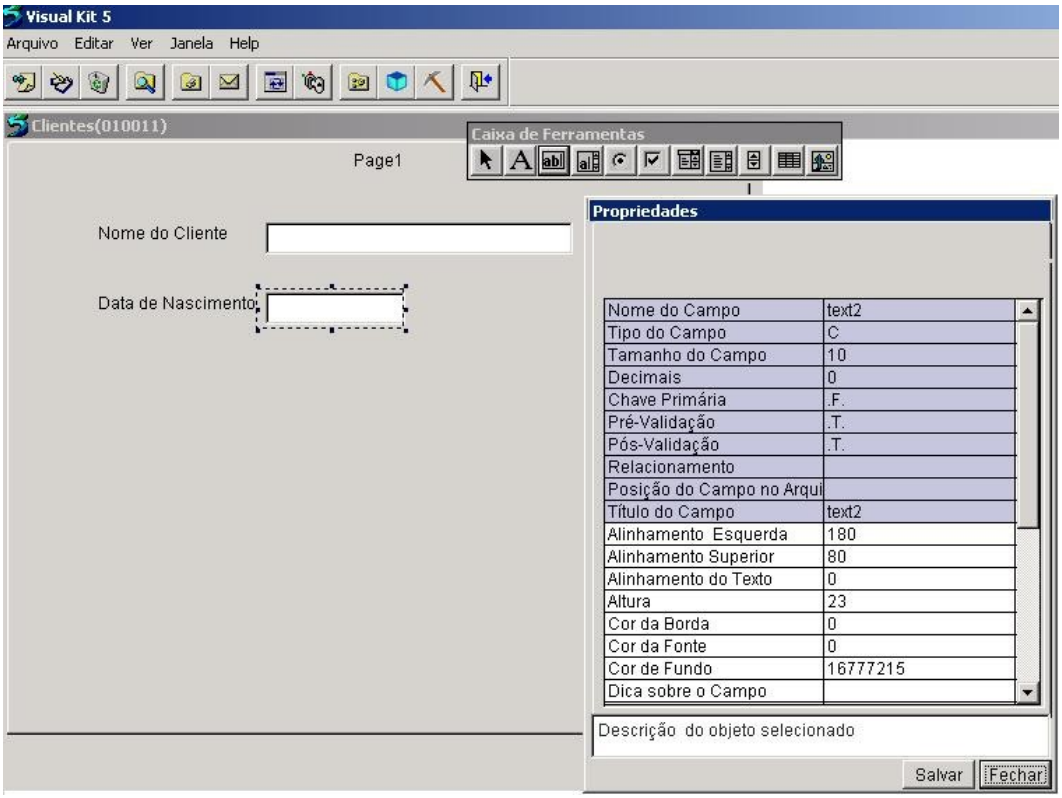


Fig 36 Desenhando o formulário.

Salvando o sistema e fechando o desenho

Para salvar pressione o botão **Salvar** na janela “Propriedades”, e confirme a gravação do arquivo.

Uma vez salvo o arquivo, você poderá sair da janela de desenho, e sair do Visual Kit 5 para testar seu primeiro sistema . Veja os procedimentos necessários.

- Pressione o botão **Fechar** na janela de “Propriedades”.
- Aparecerá uma mensagem de confirmação *Deseja sair do sistema?:* aperte **Yes**.
- Na Janela Módulo e Entidades clique em **Cancela**.
- Aparecerá uma mensagem de confirmação *Você não selecionou uma Entidade. Deseja Selecionar uma Entidade?* aperte **No**.
- Para sair da janela "Utilize Sistemas Prontos / Cria Sistema Novo", pressione o botão **Cancela.**
- Para sair do Visual Kit 5, pressione o botão .

Acessando o sistema gerado

Vamos agora acessar o sistema que você gerou com a ajuda do Visual Kit 5 e comprovar se tudo está realmente funcionando... Para executar o sistema gerado siga as seguintes instruções:

- Dê um clique no botão Iniciar (Start) na barra de tarefas (Task Bar) do seu Windows (ou pressione CTRL+ESC).
- Selecione a opção Executar (Run);
- Na opção "Abrir" (Open), digite: **C:\vclientes\principal.exe** e clique no botão **Ok**, para executar o sistema gerado.

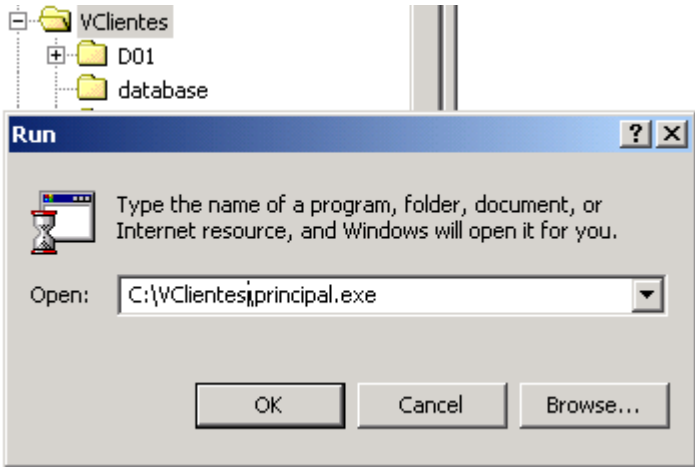


Fig 37 Executando o sistema gerado

Viu como é simples? Finalmente chegamos ao final do desenvolvimento de nosso primeiro sistema em Visual Kit 5.A Fig 38 exibe o sistema após a execução dos procedimentos anteriores. Note que é muito similar ao próprio Visual Kit 5.

Fig 38 Visual do sistema gerado pelo visual Kit 5

Testando o sistema

Bom, chegou o momento de testar o nosso sistema. Está nervoso?...Vamos ver algumas das funcionalidades do sistema que você acabou de desenvolver com ajuda do Visual Kit 5. Nos próximos capítulos colocaremos detalhadamente todos os recursos gerados.

Incluindo dados

O primeiro, é incluir dados, que é a razão de ser de qualquer sistema. Vamos incluir aproximadamente cinco nomes com suas datas de aniversários.

Exemplo:

Maria	16/12/1964
João	20/02/1980
Luiza	13/05/1990
Pedro	03/11/1974
Gabriela	10/10/1959

Para realizar esta inclusão siga os seguintes passos:

- Dê um clique duplo em cima do nome do módulo **Cadastro** ou um clique em cima do [+] associado ao módulo.
- Vai aparecer a entidade **Clientes** que você registrou.
- Dê um clique duplo em cima do nome da entidade **Clientes** ou um clique em cima do [+] associado.
- Vão aparecer as operações que o Visual Kit 5 incluiu automaticamente para esta entidade. Como tínhamos selecionado para incluir todas, foram incluídas as 12 operações básicas do Visual Kit 5..

Nota

O Visual Kit 5 inclui as seguintes operações básicas: incluir, alterar, excluir dados, fazer consultas, pesquisas, imprimir relatórios, etiquetas, criar arquivos de exportação - importar dados, criar cubos OLAP e definir modelos de Data Mining.

- Dê um clique em **Inclusão.**
- Surpresa! Apareceu o formulário que você desenhou. Repare que foram incluídas várias teclas na parte de inferior do formulário. Veja na Fig 39.
- Note também que existe um ponteiro situado no primeiro campo, sinalando de maneira intermitente, que este campo está “focado”,ou seja, neste momento ele receberá a inclusão dos dados.
- Digite então o primeiro nome de nosso exemplo e pressione ENTER.
- O cursor agora pulou e está enfocando o seguinte campo da **Data de Nascimento**. Digite a data de nascimento. Com isto, culmina a inclusão do primeiro registro da lista.
- Para salvar, você pode pressionar ENTER ou então com o mouse clicar no ícone Salvar. Imediatamente os dados serão armazenados e o sistema ficará pronto para a inclusão de novos dados.
- Terminemos de incluir os quatros registros restantes.
- Para terminar a operação de inclusão pressione a tecla ESC. Caso faça de outro jeito, originará a mensagem *Atenção: texto1 INVALIDO* . Isto é uma garantia para não gravar um valor nulo (vazio) no primeiro campo da entidade, por ser este o campo chave.

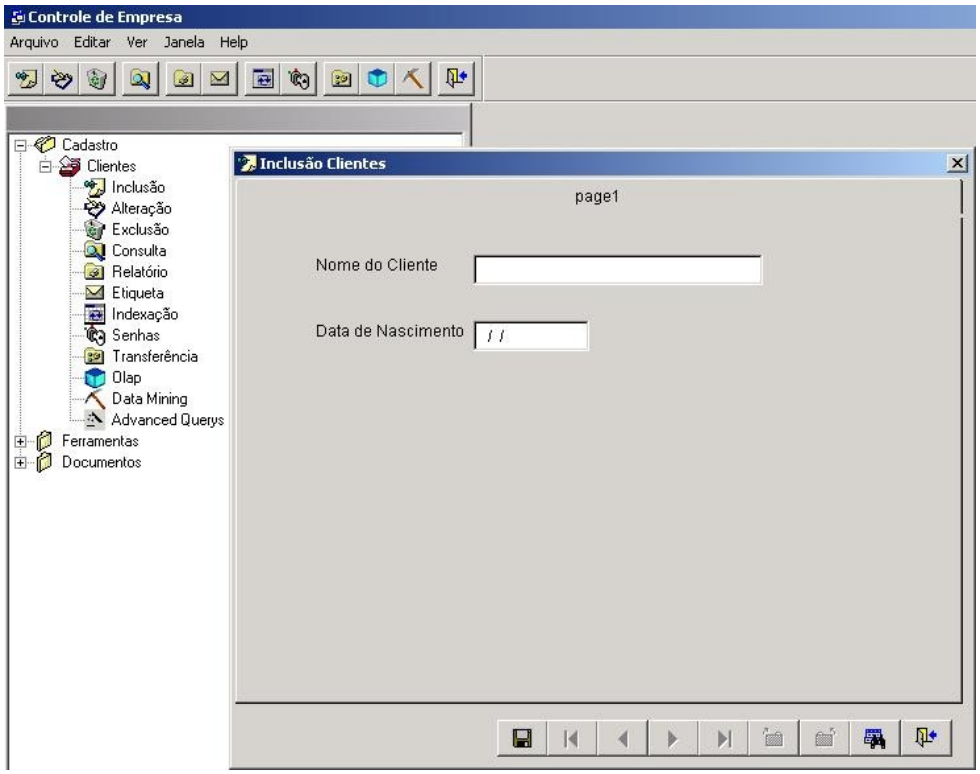


Fig 39 Visão do formulário no sistema gerado.

Depois de incluir todos os dados do exemplo, vamos tentar incluir um segundo nome de Pedro. Impossível, o sistema não permite! Está

aparecendo uma caixa de mensagem indicando "*Atenção text1 JÁ EXISTE*". Observe na Fig 40. Sabe explicar o que aconteceu?.

Certo... esse campo é chave primária, e como tínhamos falado no Capítulo II, seu conteúdo é único dentro da entidade. O Visual Kit 5 coloca por *default* como chave primária o primeiro campo da entidade, e neste caso foi o nome do Cliente (coisa não recomendável). É por isso que é considerada uma boa prática de desenvolvimento começar sempre por um campo-código que identifique o registro como um todo, como por exemplo, código do cliente, ou código do produto, etc. Assim você poderá cadastrar vários clientes com o mesmo nome e claro está – com código diferente.

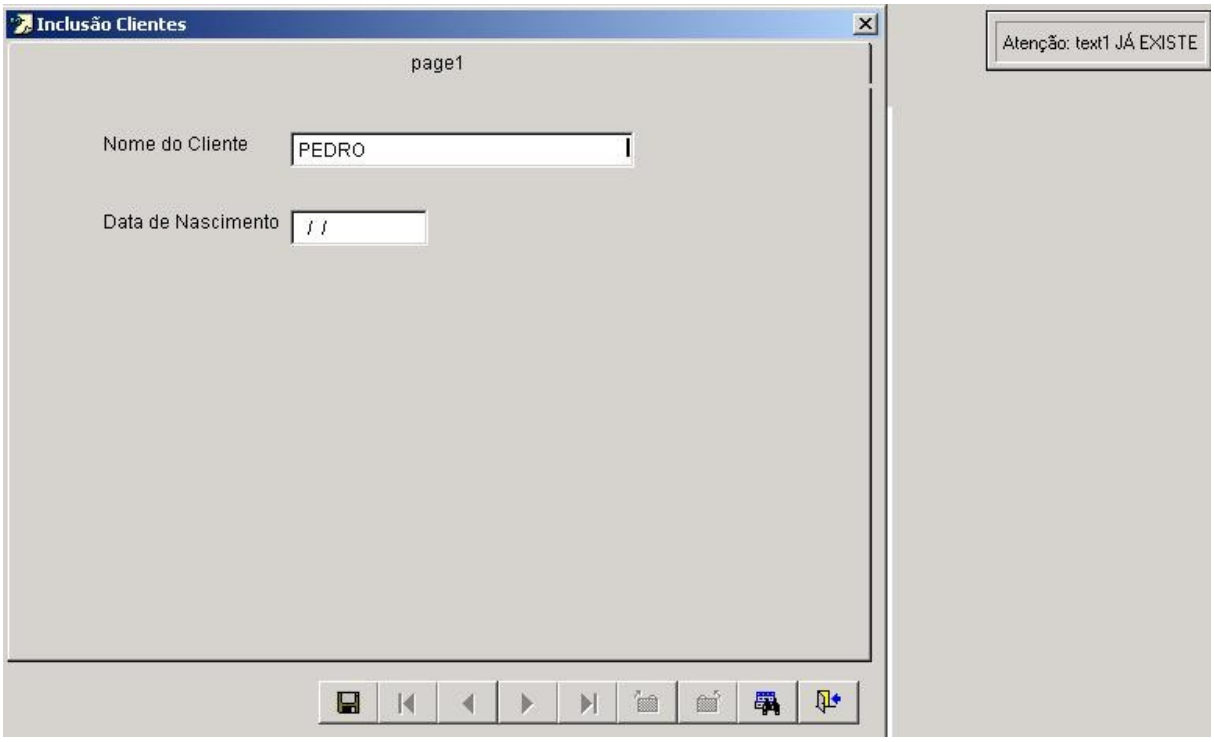


Fig 40 Cadastrando cliente com o mesmo nome.

Nota

No Visual Kit 5 o primeiro campo do formulário é sempre campo Chave e portanto seu conteúdo é único na dada entidade.

Consultando.

Usando esta operação você poderá dar consulta aos dados incluídos em formato de tabela. Lembra que falamos que todo formulário por trás contém uma tabela, onde cada coluna representa cada campo e cada linha representa um registro.

Dando um clique nesta operação aparecerá uma janela de consulta com os dados que acabamos de cadastrar.

Esta janela possui muitas opções destinadas a fazer filtros, relatórios, etc. Voltaremos no Capítulo XVII para detalhar o funcionamento de todas estas opções.

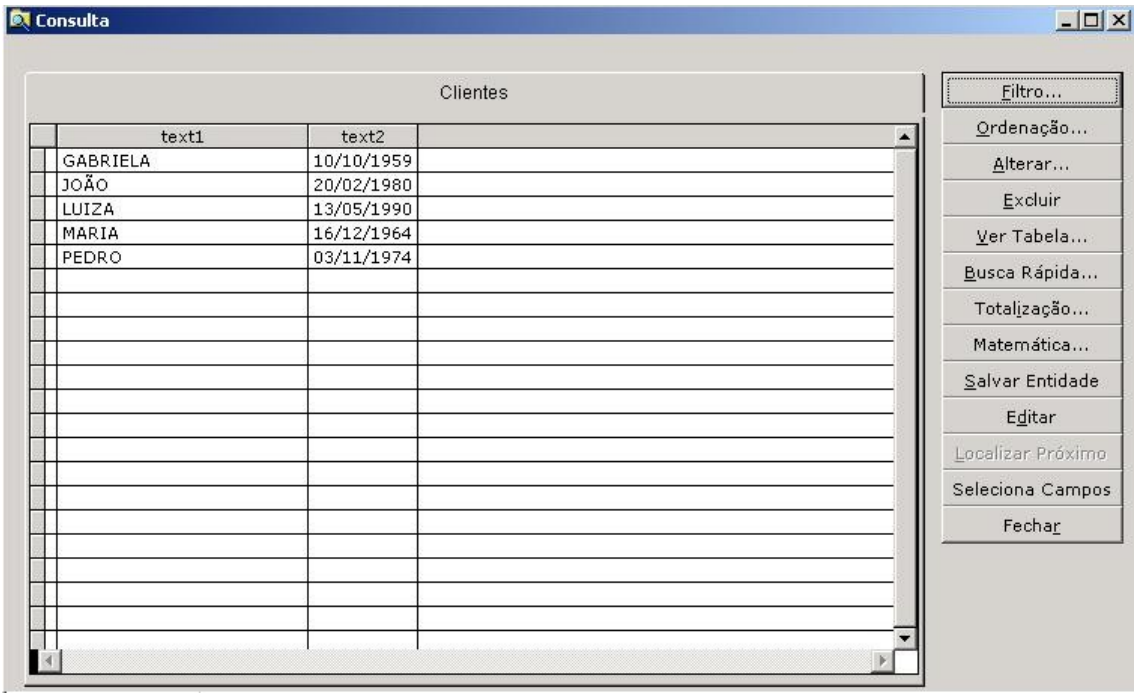


Fig 41 Consulta a dados cadastrados

Alterando dados

Esta operação permitirá exclusivamente alterar dados. Não poderá nem incluir nem apagar registros ou conteúdos dos campos. Siga os procedimentos para a alteração de dados.

- Dê um clique em **Alteração.**
- Você pode digitar um dos nomes e pressionar ENTER.
- Aparecerá a data de Nascimento referente da pessoa, a qual poderá ser alterada.
- Nesta chamada, o Visual Kit 5 não permite alterar o campo chave.
- Também você pode navegar pelos registros usando as teclas:

Exibe o primeiro registro do arquivo.

Exibe o registro anterior.

Exibe na tela o próximo registro.

Exibe o último registro do arquivo.

Excluindo dados

Esta operação permitirá apenas excluir dados. Não poderá nem incluir nem alterar registros.
Dê um clique em **Exclusão.**
Você pode digitar um dos nomes e pressionar Enter.
Aparecerá os dados da pessoa a qual poderá ser excluída clicando o botão.

Exclusão: (deleta) o registro exibido.

Modificando o sistema

Para incluir novas entidades ou alterar alguma propriedade do formulário ou até para corrigir algum erro, o sistema precisa ser reaberto novamente pelo Visual Kit 5. Todo desenvolvedor sabe que grande parte do desenvolvimento é destinado para fazer as atualizações ou modificações que o cliente ou usuário do sistema exigirá pelo resto da vida.

Reabrir o sistema é muito fácil. Veja os procedimentos necessários.

- Em primeiro lugar, você deverá abrir o Visual Kit 5.
- Clique na *TreeView* para abrir a pasta “Gerador de Sistema”
- Clique em “Sistema Completo” e selecione a opção “Inclusão”.
- Abrir-se-á uma janela, solicitando que você indique o diretório onde está o sistema a modificar. O Visual Kit 5 exibe sempre o diretório do último sistema manipulado.
- Responda que "Sim"/"Yes", confirmando que esse é realmente o diretório do sistema a modificar.
- Finalmente será aberta a janela de Módulo e Entidade, permitindo ter acesso e criar ou modificar todos os módulos e entidades do sistema.

Você deve ter notado que muitos conceitos e funcionalidades não foram ainda abordados, isto será feito nos próximos capítulos.

V. Desenhando o Formulário

A principal tarefa que o desenvolvedor Visual Kit 5 tem que realizar no desenvolvimento de sistemas é desenhar os formulários. Você deverá criar formulários que sejam amigáveis com o usuário, evitando muitos controles em um única janela e usando também cores agradáveis (imagine só um formulário com letras cor laranja no fundo vermelho!!!).

No decorrer deste capítulo, você conhecerá diversos controles que têm a mesma finalidade. Isso poderá causar dúvidas do tipo: qual deles devo utilizar? Nesses casos, opte sempre pelo controle que ofereça maior claridade ao usuário e permita a compressão da tarefa a ser executada por ele.

As ferramentas de desenho *Designer-Básica* e *Designer-Pro*

O Visual kit 5 possui duas ferramentas de desenho de formulários: a *Designer-Básica* e a *Designer –Pro*. A Designer – Básica -com menos recursos- garante a compatibilidade com versões e releases anteriores do Visual Kit 5. Já a *Designer-Pro,* possui uma serie de novos recursos, tais como *Grid* e *Botão*, diversos assistentes de criação, assim como editores de macros e códigos.

Como descrevemos no capítulo anterior, sempre que você definir uma nova entidade, o Visual Kit 5 solicitará a definição das operações básicas (veja na Fig 42). Neste momento você poderá escolher qual das ferramentas usar: a *Designer-Básica* ou a *Designer –Pro*.

Atenção!

Quando selecionamos para a dada operação a **Função Básica** estaremos acessando a *Designer-Básica*. Se selecionamos **Formulário Externo**, estaremos acessando a *Designer-Pro*.

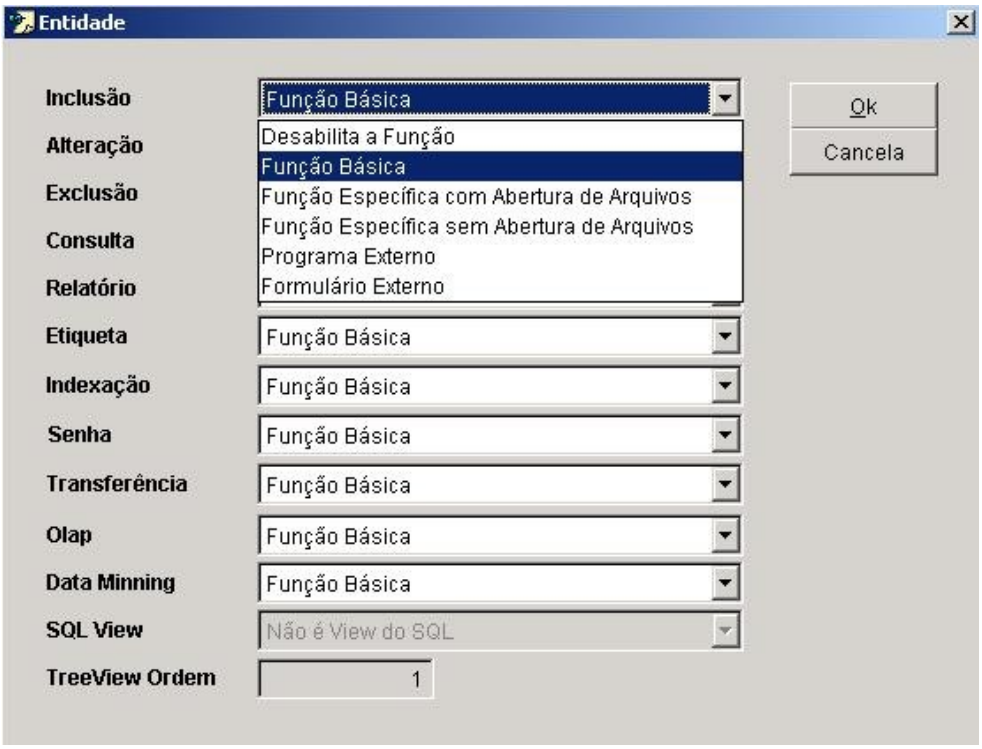


Fig 42 Escolhendo a Designer

Manipulando controles

Nesta seção, apresentaremos a caixa de controles ou caixa de ferramentas do Visual Kit 5 tanto da *Designer-Básica* como da *Designer-Pro*. Você aprenderá inserir e manipular os controles ou objetos dentro do formulário.

Conhecendo a Caixa de Ferramentas

Conforme mencionamos anteriormente a caixa de ferramentas exibe os controles que podem ser empregados para desenhar a interface dos formulários. Cada formulário poderá ter diversos controles, distintos ou não.

Cada controle indica um tipo de interface com o usuário e possui características particulares. Os controles são identificados pelo nome comumente aceito nas linguagens de programação Windows, como por exemplo *TextBox*, *Option*, *ComboBox,* etc.

Fig 43 Caixa de ferramentas da *Designer-Básica*

Fig 44 Caixa de ferramentas da *Designer-Pro*

Utilize este botão para desmarcar a seleção de objetos, na caixa de ferramentas.

Label: Criar um rótulo, usado para textos que você não deseja que o usuário altere, como por exemplo, a legenda que aparece ao lado das caixas de texto. Esta opção não cria campos na entidade.

TextBox: Cria uma caixa de texto, usado para manter uma única linha de texto a ser digitada ou alterada pelo usuário. Esta opção é a forma básica de criar campos na entidade.

EditBox: Cria uma caixa de edição, usado para manter múltiplas linhas de texto a serem digitadas ou alteradas pelo usuário. O tipo de campo que ela cria é exclusivamente do tipo Memo.

Option: Cria um grupo de opções usado para exibir múltiplas opções entre as quais o usuário pode selecionar apenas uma.

CheckBox Cria uma caixa de verificação, que permite ao usuário escolher entre uma condição verdadeira ou falsa. Cria campo tipo lógico.

ComboBox: Cria uma caixa de combinação suspensa para que o usuário possa selecionar um item da lista de item formada por uma outra entidade do sistema. Portanto, a *ComboBox* deverá obrigatoriamente ser relacionada a uma segunda entidade.

ListBox: Cria uma caixa de listagem, usado para exibir uma lista de itens de onde o usuário pode fazer uma seleção. A lista pode ser rolada se tiver um número de itens superior ao que pode ser exibido de uma só vez. O controle *ListBox* deverá obrigatoriamente ser relacionado a uma segunda entidade que contenha a lista de itens.

Spiner: Cria um controle de rotação, usado para aceitar a entrada numérica que pertence a determinado intervalo. O campo criado é sempre de tipo numérico. Utilizando o mouse pode-se somar ou subtrair 1 ao valor do campo.

Image: Exibe uma imagem gráfica (arquivo externo formato .BMP) no seu formulário.

Grid: Cria um controle Grade, usado para exibir os dados em uma grade parecida com uma planilha.

Botão: Cria um controle Botão, usado para emitir um comando.

Exibe: Controle de exibição. Usado para exibir na tela do formulário o conteúdo de um campo de outra entidade.

Adicionando controles ao formulário

A caixa de ferramentas mostrada na Fig 44 permitirá inserir ou desenhar os controles nos formulários. Está é uma manipulação fundamental que você deverá aprender para poder criar sistemas com o Visual Kit 5. Para adicionar:

- Selecione primeiro o controle desejado na caixa de ferramentas fazendo um clique no ícone correspondente
- Leve o ponteiro do mouse para o local desejado dentro do formulário.
- De um clique no mouse e o controle será desenhado no lugar indicado.

Selecionando controles

Sempre que tiver que alterar alguma característica relativa a algum controle, antes de executar os comandos ou procedimentos necessários para tal, você deverá selecioná-lo.

Para selecionar um único controle dê um clique no objeto desejado dentro do formulário. Caso o objeto desejado seja o próprio formulário, dê um clique em uma área em que não haja controles no mesmo.

Visualmente, um controle selecionado apresenta quadrados em sua volta. Tais quadrados são denominados de “marca de seleção”. Veja na seguinte figura.

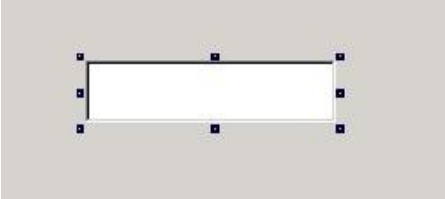


Fig 45 Marcas de seleção

Para selecionar vários controles dentro do formulário você pode fazer de duas maneiras. A primeira é da seguinte forma:

- Mantenha a tecla SHIFT pressionada;
- Dê um clique em cada um dos objetos a serem selecionados. Note que cada controle que faz parte da seleção aparece rodeado de pequenos quadros negros.

Outra forma muito eficaz é utilizando o ponteiro do mouse na caixa de ferramentas e arrastá-lo emoldurando todos os controles a serem selecionados. Os controles que estiverem com alguma parte visível dentro da moldura serão selecionados. Veja na seguinte figura.

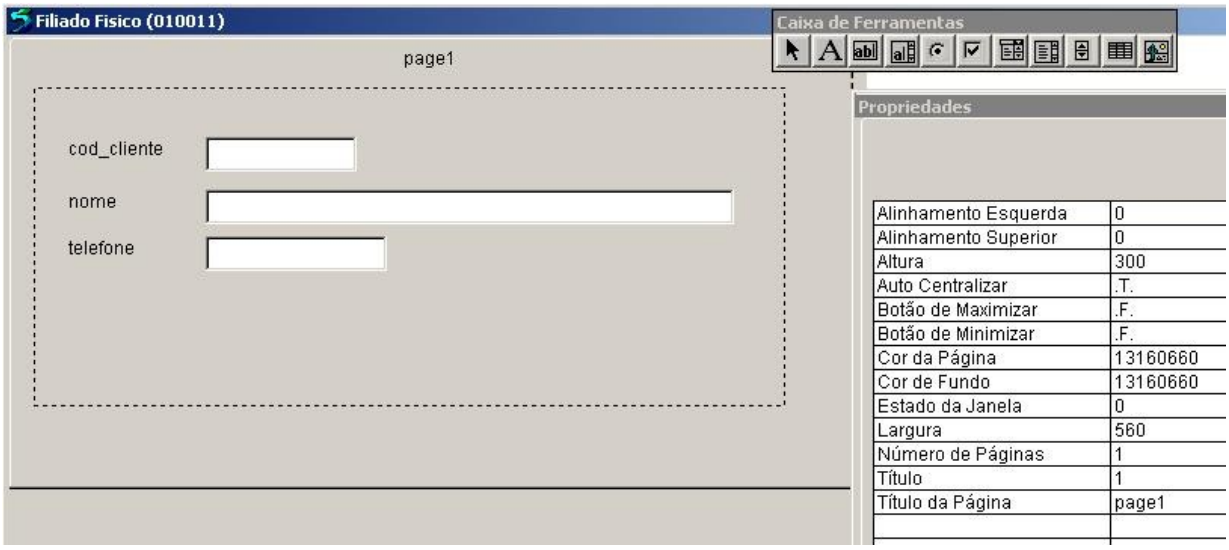


Fig 46 Selecionando vários objetos com o mouse.

Quando vários controles são selecionados, a janela de propriedades mostrará somente as propriedades comuns a todos, como mostra a Fig 47.

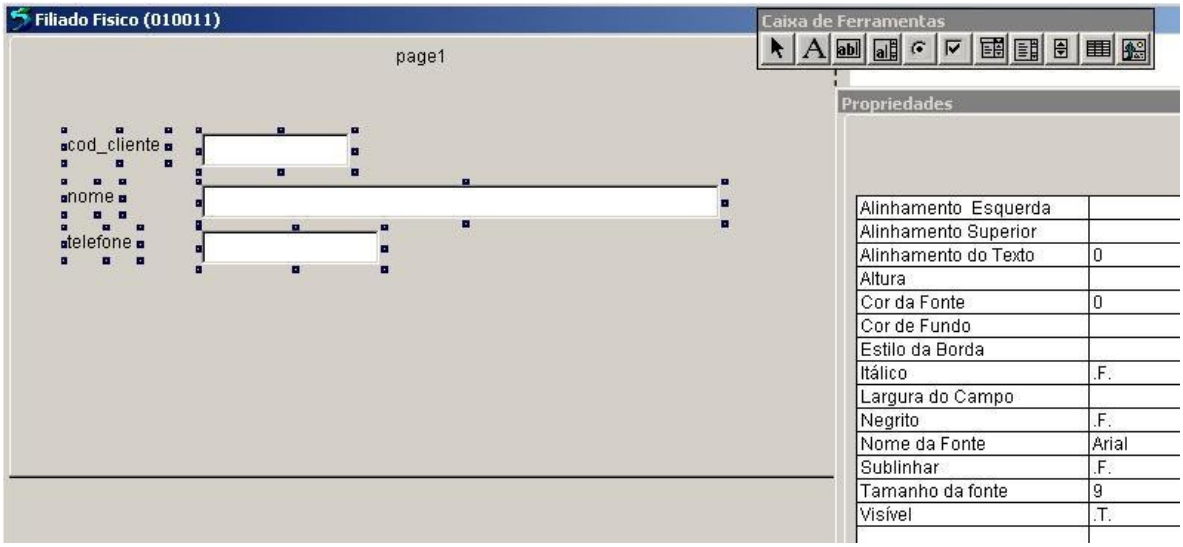


Fig 47 Controles selecionados.

Dica

Sempre que um controle é inserido no formulário, este automaticamente já estará selecionado.

Posicionando controles

Após compreender a seleção dos controles, outra tarefa relevante para o desenho do formulário é posicioná-los dentro da própria janela do formulário na posição desejada. Por meio da seleção e posicionamento dos controles, você poderá desenhar sua interface de maneira que a mesma atenda justamente às suas necessidades.

Ao mover um controle, automaticamente você estará modificando os valores para as propriedades **Alinhamento Superior** e **Alinhamento Esquerda** do mesmo.

Para mover os controles você poderá usar tanto o mouse como o teclado:

Usando o mouse:

- Selecione o controle desejado;
- Dê um clique no centro do controle e arraste o mouse para a posição desejada. No caso de seleção de vários controles, dê um clique e arraste qualquer um deles e todos serão movidos.

Usando o teclado:

- Selecione o controle desejado;
- Utiliza as teclas de direção correspondente ao sentido em que você quer mover o controle.

Dimensionando controles

Esta tarefa será necessária sempre que você quiser desenhar controles com tamanhos diferentes do *default*.. Os procedimentos são:

- Selecione o controle desejado;
- Coloque o ponteiro do mouse sobre uma das marcas de seleção (um dos quadros negros que emolduram o objeto), conforme sentido desejado para a nova dimensão;
- Arraste o mouse, soltando o botão quando a moldura atingir a dimensão desejada.

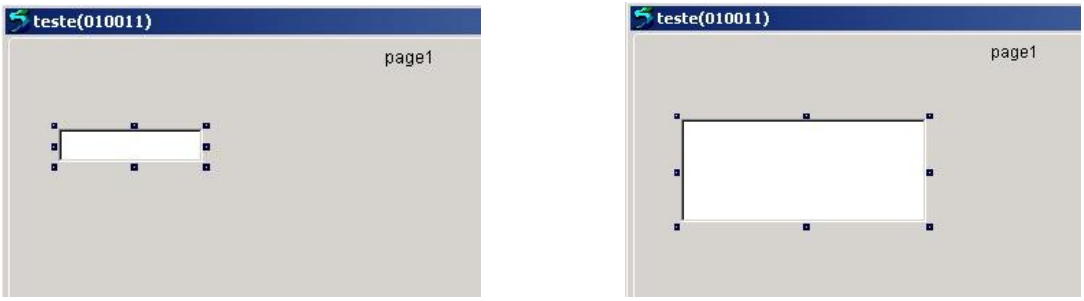


Fig 48 Dimensionando controles (Antes e Depois)

Para dimensionar um controle usando o teclado, faça o seguinte:

- Selecione o controle a ser dimensionado;
- Mantenha a tecla SHIFT pressionada e pressione a tecla de direção correspondente ao sentido desejado.

Excluindo controles

Para remover um ou mais controles do formulário, faça desta maneira:

- Selecione o(s) controles desejado(s).
- Pressione a tecla DEL (*Delete*).

Copiando, Recortando e Colando controles

Existe uma forma de copiar, recortar e colar objetos dentro do formulário. Os procedimentos seguem o padrão Windows: você seleciona o controle que deseja copiar/cortar e depois executa o comando de colagem na posição onde deverá ser inserido:

- Selecione o controle que deseja copiar e/ou recortar;
- Acione o menu de contextos dando um clique com o botão direito do mouse;
- Selecione a opção *Copiar* ou a opção *Cortar*.

Para colar os objetos copiados, acionamos novamente o menu de contexto e selecionamos a opção *Colar*.

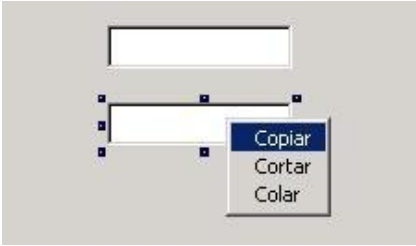


Fig 49 Menu de contextos.

Alinhamento de controles

Às vezes, posicionar vários controles alinhados resulta uma tarefa difícil de realizar manualmente ou simplesmente chata. Você poderá alinhar os controles de uma maneira rápida e fácil da seguinte forma:

- Acione o menu de contextos dando um clique com o botão direito do mouse em uma área do formulário em que não haja controles no mesmo;
- Selecione a opção *Ajustar Campos* no menu de contextos. E pronto!

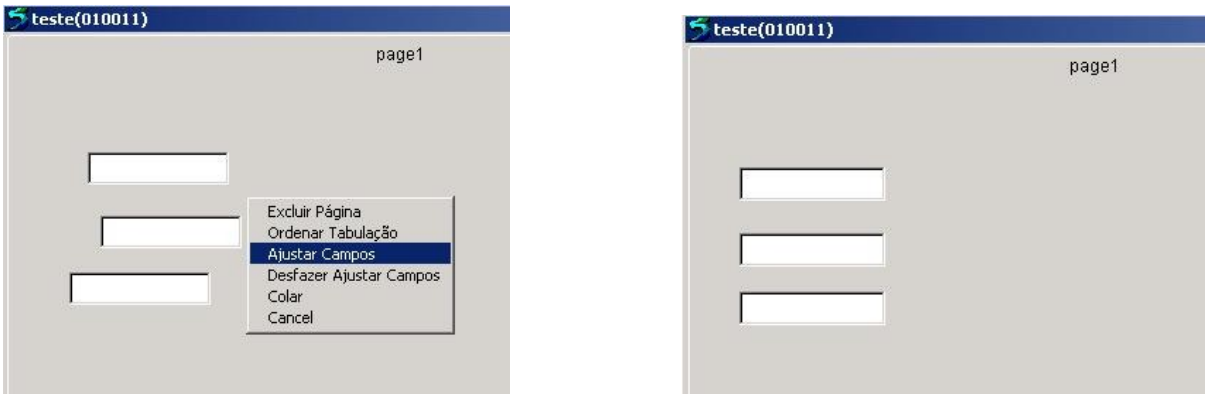


Fig 50 Antes e depois de ajustar os controles.

Definindo ordem de tabulação

A ordem de tabulação dos campos em um formulário é a ordem em que os objetos são selecionados ou “focados” quando o formulário já está em funcionamento. *Foco* é o termo que usamos para descrever o campo que

detém o controle no momento. Quando um campo recebe um clique ou é ativado por meio da tecla TAB, ele ganha o foco.

Nesta fase de desenho, você pode decidir a ordem de tabulação dos controles, que determinará por sua vez a ordem em que será feita a entrada de dados no sistema quando o formulário entre em funcionamento.

- Dê um clique com o botão direto do mouse em uma área do formulário em que não haja controles no mesmo para acionar o menu de contextos;
- Com o ponteiro do mouse selecione *Ordenar Tabulação* na janela.;
- Ao selecionar a Ordem de Tabulação serão apresentadas umas caixinhas com os números correspondentes à ordem de acesso aos controles, conforme mostra a Fig 51;
- Clique várias vezes em cima do número até que fique na ordem desejada.
- Pule para a próxima caixinha que automaticamente a anterior ficará fixada – ficando de cor cinzenta.

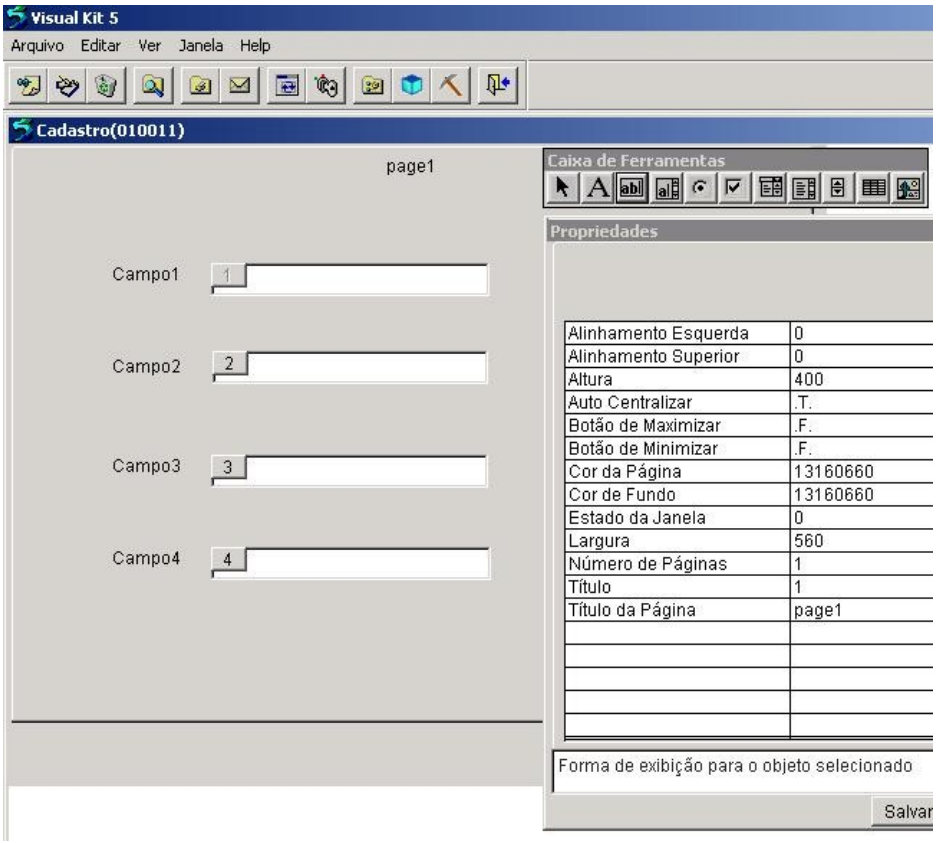


Fig 51 Ordem de tabulação.

Nota

Os campos chaves recebem o primeiro foco.
Os controles ou objetos que nunca ganham foco como o *Label* não precisam da ordem de tabulação.

A janela de propriedades é uma janela fundamental no desenvolvimento dos seus formulários, já que por meio dela você personalizará tanto o visual dos controles como o funcionamento dos mesmos.

As “propriedades” simplesmente são as características especifícas de cada controle. A intenção desta janela é concentrar num lugar só todas as propriedades relativas ao controles, e assim facilitar o acesso e manuseio das mesmas.

Em resumo, desenhar o formulário consiste nos seguintes passos:
- Inserir os controles nos formulários
- Distribuí-los e dimensioná-los de forma clara e objetiva.
- Alterar os valores de suas propriedades para personalizá-los, incluindo também certas regras de negócio.

Entendendo as Propriedades

Dois elementos formam parte de qualquer propriedade, o *nome* da propriedade, que identifica a característica a ser manipulada, e o *valor* que especifica seu conteúdo. Por exemplo:

Nome	Valor
Altura	400
Largura	560
Alinhamento Esquerda	50
Alinhamento Direita	20

Cada tipo de controle tem um conjunto de propriedades que possuem um valor inicial ou *default*. O que é um valor *default*?. Se você simplesmente inserir vários controles no formulário conforme estudamos anteriormente sem alterar nenhum valor das propriedades dos formulários, salvar o sistema e rodar ele, o mesmo apresentará algumas características, como

por exemplo a cor cinza do fundo do formulário, pois bem, tais características são os valores *default* das propriedades.

A maioria das propriedades pode ficar *default*, e você terá que ajustar e definir apenas os poucos que interessam ao controle sendo definido.

Manipulando a Janela de Propriedades

Para acionar a janela de propriedades, basta selecionar o controle do qual você deseja visualizar as propriedades que automaticamente a janela de propriedades assumirá os valores do controle selecionado. O conteúdo da janela de propriedades poderá variar com o tipo de controle selecionado, pois cada controle possui um conjunto de propriedades particulares.

Caso você selecione vários controles de maneira conjunta, aparecerá exclusivamente as propriedades comuns dos controles selecionados.

Observe a Fig 52, vamos localizar as áreas disponíveis na janela de propriedades. Note que existem propriedades com um fundo azulado e outras com fundo branco. As propriedades com fundo azulado são as relacionadas ao banco de dados e as propriedades de fundo branco são as relacionada estritamente com a estética ou parte visual do formulário.

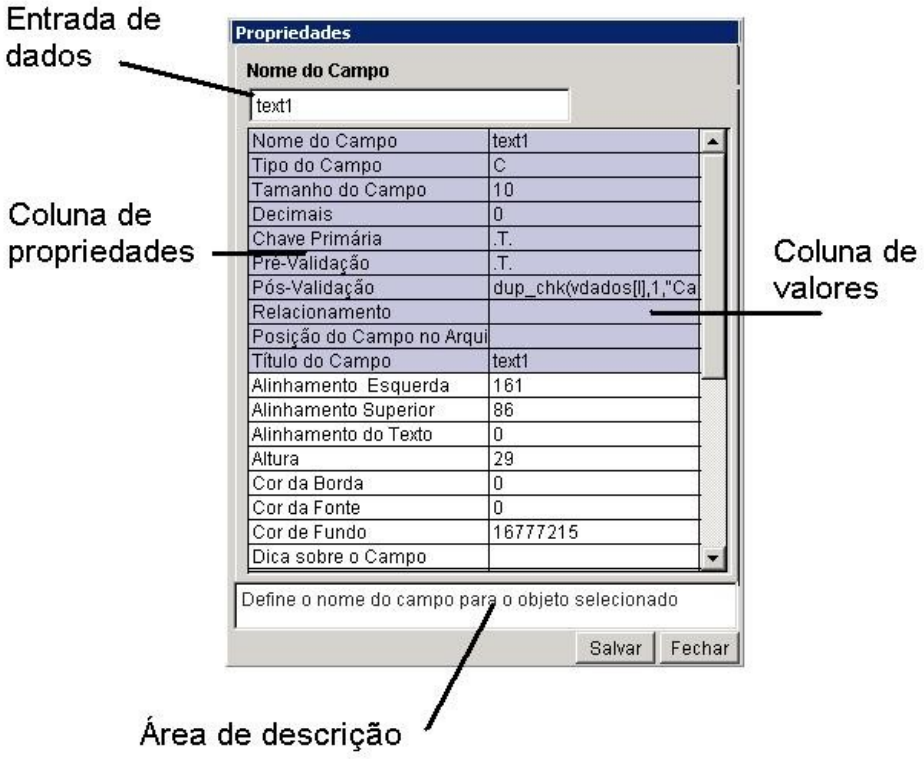


Fig 52 Apresentação da janela de propriedades

Coluna de Propriedades

A janela de propriedades está dividida em duas colunas. A primeira lista o nome de todas as propriedades disponível para o controle ou conjunto de controles selecionados.

Coluna de Valores

Esta coluna apresenta o valor atual da propriedade imediatamente à sua esquerda. A alteração da propriedade pode se executada de diferentes formas:

Caso o valor da propriedade seja uma seqüência de caracteres (*string)*. Como exemplo vamos mudar o valor *default* do título da página do formulário da seguinte maneira.

- Selecione a entidade
- Selecione a janela do próprio formulário - dê um clique em uma área em que não haja controles no mesmo.
- Dê um clique na propriedade **Título da Página.** Note que aparecerá uma entrada de dados na parte superior da janela de propriedades.
- Digite o novo valor, por exemplo *Testando.*
- Clique no botão *Salvar* e confirme; ou clique em outra propriedade ou selecione outro controle do formulário.

Algumas propriedades possuem valores preestabelecidos. Neste caso, você poderá alterá-los selecionando a propriedade de uma lista apresentada. Veja a seguir o exemplo de alteração deste tipo de propriedade.

- Selecione a entidade
- Selecione um controle *TextBox* do formulário.
- Dê um clique na propriedade **Tipo de Campo.** Note que aparecerá uma *ComboBox* (você já sabe que é uma *ComboBox*) com a lista das propriedades na parte superior da janela de propriedades.
- Clique no novo valor, por exemplo campo tipo *N- Numérico* e pronto.

Existem também propriedades de manipulação de cores. Nestes casos é apresentada uma “paleta” de cores, basta dar um clique na cor desejada e pronto. Veja no exemplo:

- Selecione o próprio formulário e dê um clique em uma área em que não haja controles no mesmo.
- Dê um clique na propriedade **Cor de Fundo**. Esta propriedade altera a cor de fundo dos controles.

- Note que aparecerá um botão com reticências na entrada de dados. Clique sobre ele e será apresentada uma “paleta” de cores.
- Dê um clique na cor desejada e pronto.

Área de Descrição

Para cada elemento listado na janela de propriedades que você selecionar, a descrição é mostrada na parte inferior da janela. Note também que usar a janela de propriedades é um eficiente meio de aprender usando.

VI. A *Designer-Básica*

Principais controles

Como vimos anteriormente, o Visual Kit 5 traz consigo um conjunto de controles ou objetos. Você pode ser muito criativo utilizando o mesmo controle para diferentes finalidades, no entanto, cada um deles possui uma representatividade específica estabelecida nas aplicações Windows.

Nesta seção, você verá as finalidades de cada controle e as propriedades inerentes a ele. Como você deve haver notado, algumas propriedades estão disponíveis para vários tipos de controles, então para não ser repetitivo, comentaremos tais propriedades uma única vez.

Form – Formulário

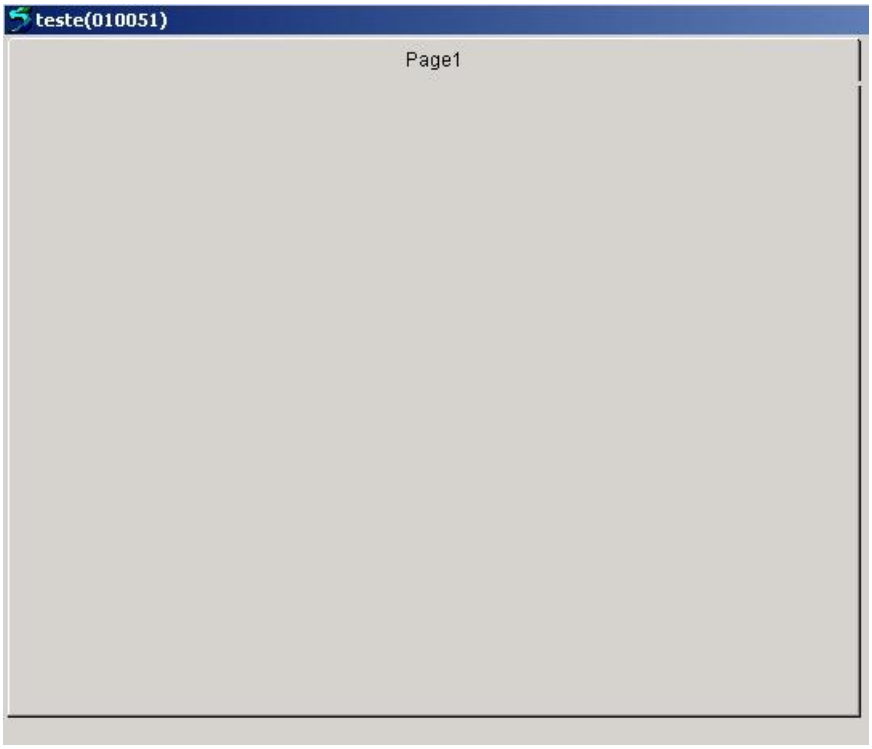


Fig 53 Visual do objeto *Form* - tela do formulário

A tela do formulário, também chamada de objeto *Form*, é basicamente o principal objeto da interface criada pelo Visual Kit 5. Este objeto permitirá

que você insira outros objetos nele com a finalidade de criar as telas de sua aplicação. Vejamos na seguinte tabela algumas das propriedades do *Form*.

PROPRIEDADE	DESCRIÇÃO
Altura	Permite definir a altura do controle
Largura	Similar a propriedade anterior, permite definir a largura do objeto.
Cor da Página	Define a cor da página do formulário.. Para a mudar a cor, localize o botão com reticências. Clique sobre ele e será apresentada uma "paleta" de cores padrão Windows para seleção -veja na Fig 54. Selecione uma cor e pronto – veja na Fig 55.
Cor de Fundo	Define a cor nas áreas fora da página. Procedimento idêntico ao anterior.
Número de Páginas	Define número de páginas dentro do formulário. Veja na Fig 56. Esta propriedade é muito útil quando o formulário possui muitos campos que não cabem numa tela só.

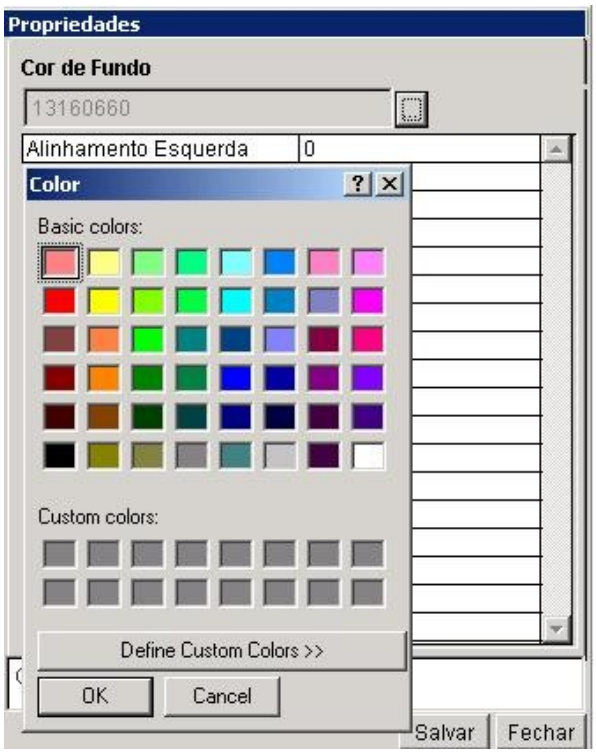


Fig 54 Mudando a cor da pagina do formulário.

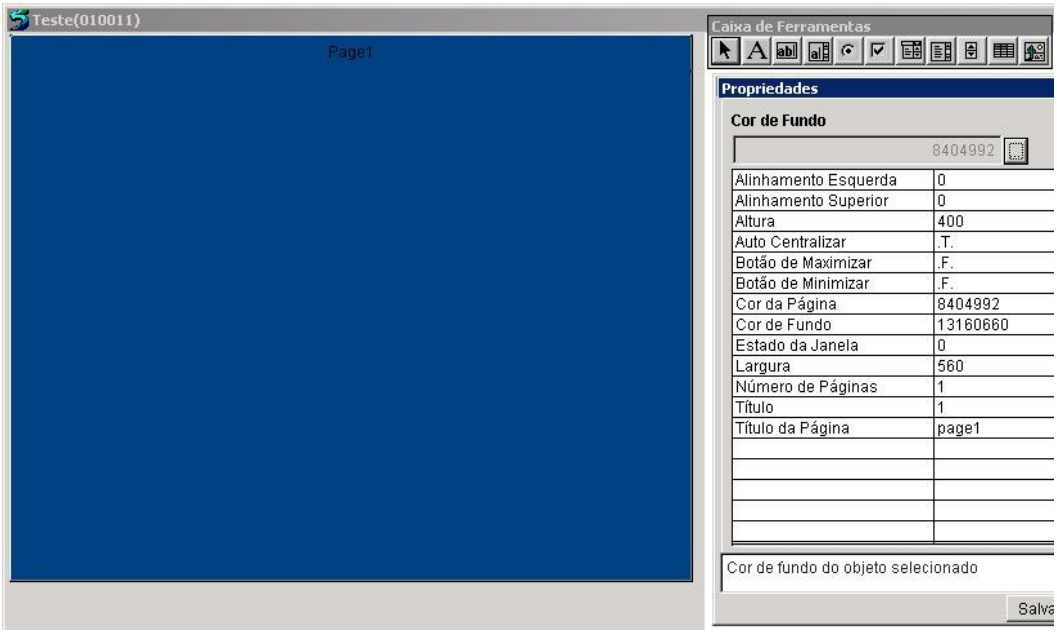


Fig 55 Formulário com nova cor de pagina

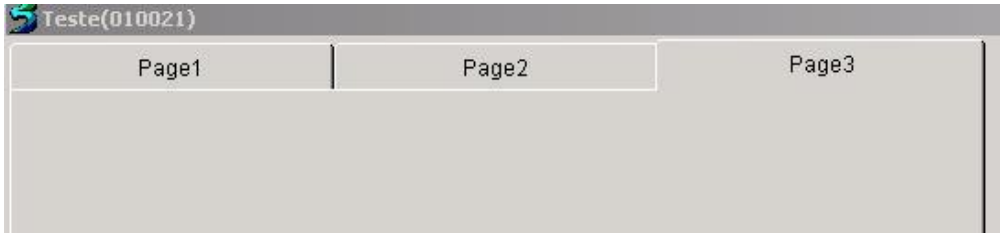


Fig 56 Formulário com três paginas.

TextBox.

Fig 57 Visual da *TextBox*

Este é um controle simples, mas fundamental no desenvolvimento de sistemas. Podemos afirmar que o *TextBox* é um pequeno editor de textos (numa única linha) que por meio dele criamos as entradas de dados básicas dos formulários.

Quando você insere um *TextBox* no formulário, O Visual Kit 5 fornece os principais recursos de texto para esse controle. Você poderá copiar,

recortar e selecionar textos usando as teclas padrão do Windows. Na tabela veja algumas das propriedades disponíveis para este controle.

PROPRIEDADE	DESCRIÇÃO
Nome do Campo	Esta propriedade é uma das mais importantes e está disponível para todos os controles do Visual Kit 5. Por meio dela você poderá atribuir um nome que identificará o controle dentro do formulário. Nas tabelas associadas aos formulários, os nomes das colunas vão assumir o nome desta propriedade. Por *default*, cada controle possui um nome, geralmente o mesmo texto da propriedade Título do Campo. Número máximo de caracteres para o nome é dez (10)
Tipo do Campo	Define o tipo de dado que o campo armazenará. O Visual Kit 5 preestabelece a seleção dos valores desta propriedade. C-Caráter: Qualquer texto. N-Numérico: Números inteiros ou frações. Intervalo de -.99999999999E+19 até .9999999999E+20 L-Lógico: Valor booleano. Verdadeiro (T) ou Falso (F). O valor deste campo pode ser mudado usando a propriedade de Pós-Validação como veremos adiante. M-Memo: Campo de tamanho variado. Refere-se a um bloco de dados. G-Geral: (foto ou objeto OLE): D-Data: data cronológica constituída de dia, mês e ano. Intervalo 01/01/100 até 12/31/9999
Tamanho do Campo	Especifica o número máximo de caracteres que o TextBox poderá receber. Caso o Tipo de Campo seja numérico a vírgula decimal contará como caracter.
Decimais	Especifica o número de dígitos à direita da vírgula decimal. Propriedade ativa somente quando o Tipo de Campo é numérico.
Chave Primária	Especifica se o campo associado ao controle é chave primária. Verdadeira – chave primária ou Falso- não é chave.
Pré-Validação	Não é estritamente uma propriedade e sim uma função que será executada antes do campo receber o foco. Estas funções serão debatidas em capítulo a parte.

Pós-Validação	Não é estritamente uma propriedade e sim uma função que será executada após o campo perder o foco. Esta propriedade será descrita em capítulo a parte.
Relacionamento	Especifica com qual entidade este campo estará relacionado. Neste caso ele receberá as propriedades do campo chave da entidade relacionada. Durante o funcionamento o *TextBox* poderá receber unicamente valores incluídos previamente no campo chave da entidade relacionada.
Posição do Campo no Arquivo	Especifica a ordem interna (não visual) do controle dentro do formulário. Coincide com a ordem do campo a coluna dentro da tabela associada ao formulário.
Título do Campo	Descrição do campo. Será a maneira de como o campo aparecerá nas operações de consultas e relatórios. Não tem limite de caracter, mas procure sempre ser breve nas definições.
Formato do Campo	Especifica o formato em que será recebido e guardado o conteúdo do *TextBox*.. Ver no Apêndice
Máscara de Edição	Especifica a máscara ou a maneira que o conteúdo será exibido na tela do computador ou em relatórios. Ver no Apêndice
Formato de senha	Especifica um caracter que deverá ser apresentado sempre que houver uma digitação dentro do *TextBox*. Independente do valor especificado para esta propriedade sempre será armazenado o valor real digitado pelo usuário.
Nome da Fonte	Esta propriedade permite alterar características de letras que podem ser utilizadas para o texto em questão. Para alterar o valor desta propriedade, dê um clique nela; isto exibirá uma caixa de seleção para selecionar a fonte.

ListBox

Fig 58 Visual do *ListBox*

O controle cria uma caixa de listagem, onde exibe uma lista de itens para que o usuário possa selecioná-los. Você deverá utilizar o *ListBox* sempre

que tiver de fornecer várias opções ao usuário, como por exemplo, uma lista de profissões dentro de um cadastro de clientes. A lista pode ser rolada se tiver um número de itens superior ao que pode ser exibido de uma só vez.

PROPRIEDADE	DESCRIÇÃO
Relacionamento	Identifica a entidade a que estará relacionado ao controle. Na entrada de dados será mostrada os nomes das tabelas segundo o padrão Visual Kit 5.
Dica sobre o campo	Insira aqui o texto que será apresentado em uma caixa de texto amarelada (conhecida como *ToolTip*), quando o ponteiro do mouse fica sobre o controle por alguns segundos

O controle *ListBox* deverá obrigatoriamente estar relacionado a uma segunda entidade que gerenciará e armazenará a lista de itens. Esta entidade deverá contar com no mínimo dois campos: o campo chave primária da lista - por exemplo o código da profissão- , mais um outro campo, - por exemplo nome da profissão. Na lista do controle *ListBox* aparecerá sempre o segundo campo da entidade relacionada, mas gravará na entidade atual o campo chave da entidade relacionada. Para maior compressão implementaremos o seguinte exemplo.

Exemplo de sistema com entidades relacionadas.

A proposta –lembre-se sempre de fazer a proposta antes de começar a desenvolver qualquer sistema: Vamos criar um sistema composto por um módulo e duas entidades. O módulo *Cadastro* e as entidades *Clientes* e *Profissão*. O objetivo da entidade *Clientes* será gerenciar diferentes informações referentes a clientes, por exemplo: nome completo, profissão, sexo, etc. Como exigência, o conteúdo do campo profissão deve constar numa lista pré-estabelecida de profissões. O objetivo da entidade *Profissão* então será precisamente gerenciar esta lista.

- Crie um módulo e as duas entidades da mesma maneira explicada no capítulo IV

Fig 59 Criamos o Módulo *Cadastro* com as Entidades *Clientes* e *Profissão*

Desenhemos primeiramente o formulário da entidade *Profissão*.

- Insira dois controles *TextBox* com seus respectivos rótulos (*Label*) identificativos.
- O primeiro campo, será o código da profissão e é campo chave primária. Mude a propriedade **Nome do Campo** para *CodProf* por exemplo e o **Título do Campo** para Código da Profissão.

Fig 60 Formulário da entidade Profissão

Atenção: É importante mudar a propriedade **Nome do Campo** dos campos chaves. Se deixar os valores *default*, ao ser relacionado o campo em outra entidade, pode ser que coincida com algum outro nome default dessa entidade, aparecendo dois campos com o mesmo nome. Isto causará um erro na execução. Recomendamos colocar nomes para cada campo que o identifiquem segundo o tipo de informação que vai gerenciar.

- Desenhemos agora o formulário *Clientes*. Coloquemos os controles *TextBox* para os campos Cód. do Cliente e Nome.

- Insira um controle *ListBox*. Este campo estará relacionado com a entidade *Profissão,* portanto, é preciso definir o relacionamento.

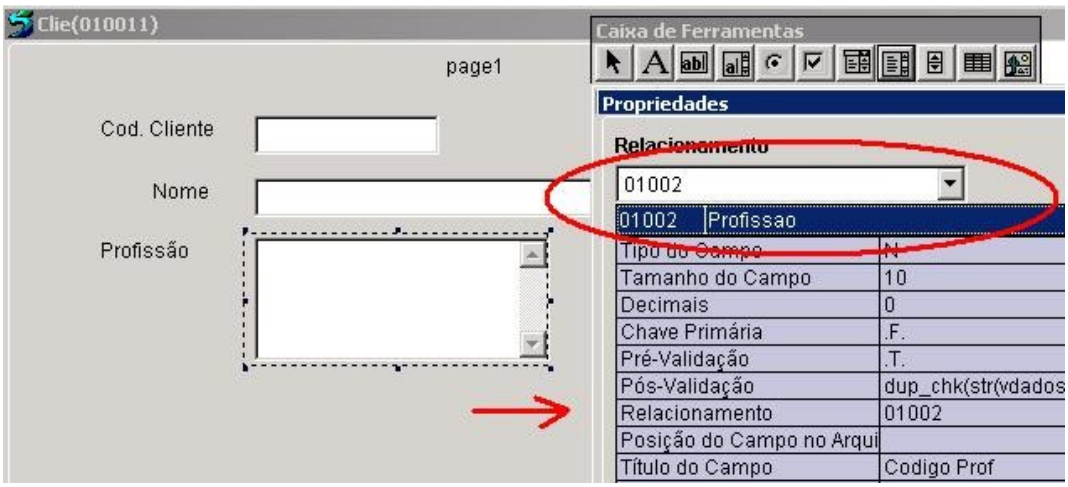


Fig 61 Formulário da Entidade Clientes

Para inserir a propriedade de relacionamento:

- Dê um clique sobre a propriedade **Relacionamento**. Aparecerá uma entrada de seleção com as tabelas do sistema (neste caso a entidade *Profissão* ou **01002**- segundo padrão Visual Kit 5, veja no Apêndice).
- Dê um clique sobre ela e será relacionada.
- Salve agora o sistema. Antes de fechar a seção de desenho aparecerá a janela de tabelas. Dê um clique no botão OK.

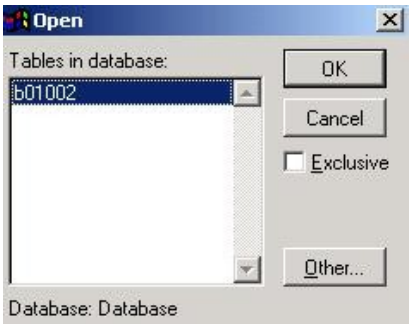


Fig 62 Fechando a seção de desenho de formulário

Vejamos como ficou nosso sistema relacionado. Vamos executá-lo como foi explicado no capítulo IV.

Primeiramente, escolhemos a entidade *Profissão* para cadastrar várias profissões. Cadastre várias profissões. Na Fig 90 aparece a consulta de algumas profissões cadastradas como no exemplo.

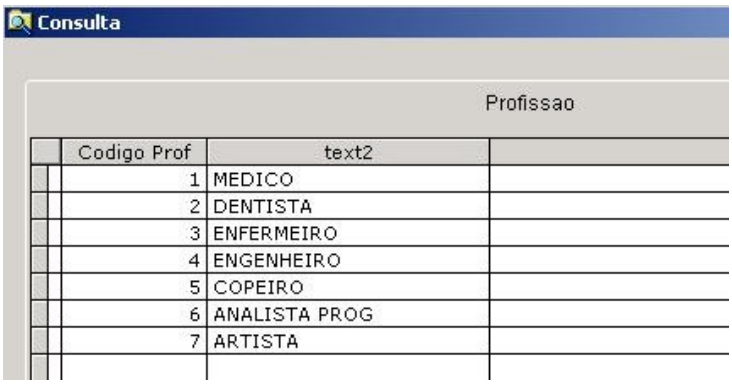


Fig 63 Profissões cadastradas.

Vamos incluir agora diferentes clientes e ver como funciona o relacionamento entre as entidades. O formulário dos clientes ficou como mostra a Fig 91. Ele possui uma lista para seleção, com as profissões cadastradas no passo anterior...Bonito eh!...

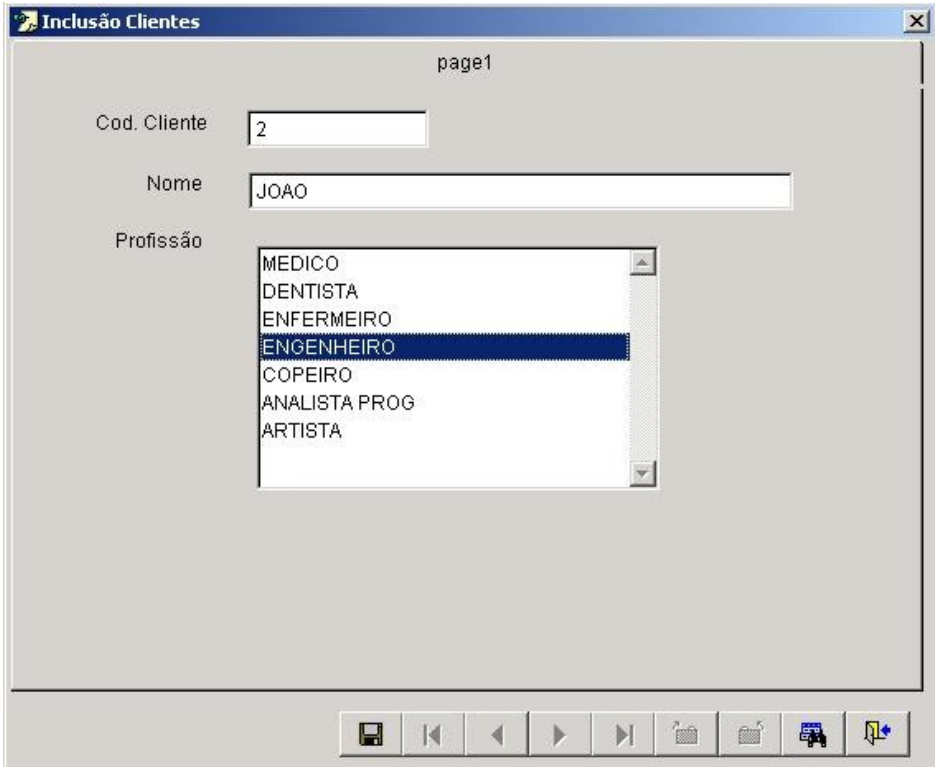


Fig 64 Formulários de Clientes com *ListBox*

Cadastramos vários clientes, use nome de amigos, namorada, familiares, etc... Vamos dar uma olhada usando a operação de *Consulta* e ver como estão sendo gravados os dados?. A primeira e a segunda coluna gravarão o código e o nome do cliente respectivamente, já a terceira coluna ficou com o código da chave das profissões. Veja na Fig 92.

Fig 65 Consulta da Entidade *Clientes*.

ComboBox:

Fig 66 Visual do controle *ComboBox*

O *ComboBox* combina uma caixa de texto (*TextBox*) com uma listagem de opções (*ListBox*). Como o *ListBox*, o *ComboBox* permite exibir uma lista de opções para o usuário, mas com a seguinte diferença:

- No *ListBox* o usuário deverá sempre dar um clique no item desejado e, no *ComboBox*, o usuário poderá tanto selecioná-lo quanto digitá-lo no *TextBox* por ele apresentado.
- Um *ListBox* sempre apresentará uma porção da lista, estará sempre aberto ocupando mais espaço no formulário. O *ComboBox* poderá ser apresentado fechado.
- Você deverá utilizar o *ComboBox* sempre que quiser exibir uma lista de opções ao usuário e não houver muito espaço livre em sua interface, ou caso você deseje oferecer duas formas de o usuário especificar o item desejado: selecionado-o ou digitando-o.

O *ComboBox* igual que o *ListBox* precisa de uma outra entidade que controlará e armazenará a listagem das opções. Por tanto, é um controle que precisa estar relacionado com outra entidade.

Para ver como funciona, vamos criar um outro sistema idêntico ao anterior, substituindo o controle *ListBox* pelo *ComboBox* . Tudo é feito da mesma maneira, para tanto vamos pular a descrição, mas não esqueça de colocar a propriedade de relacionamento.

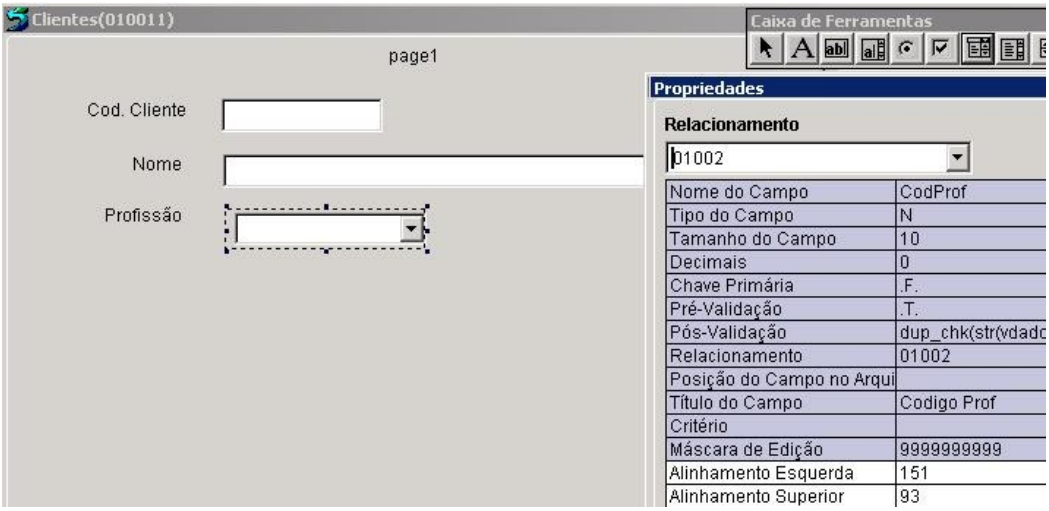


Fig 67 Colocando o controle *ComboBox*

Vejamos como funciona este controle:

- Você clica no controle e aparece a lista de profissões, veja na Fig 96;
- Selecione a profissão;
- E o controle é fechado novamente como mostra a Fig 97.

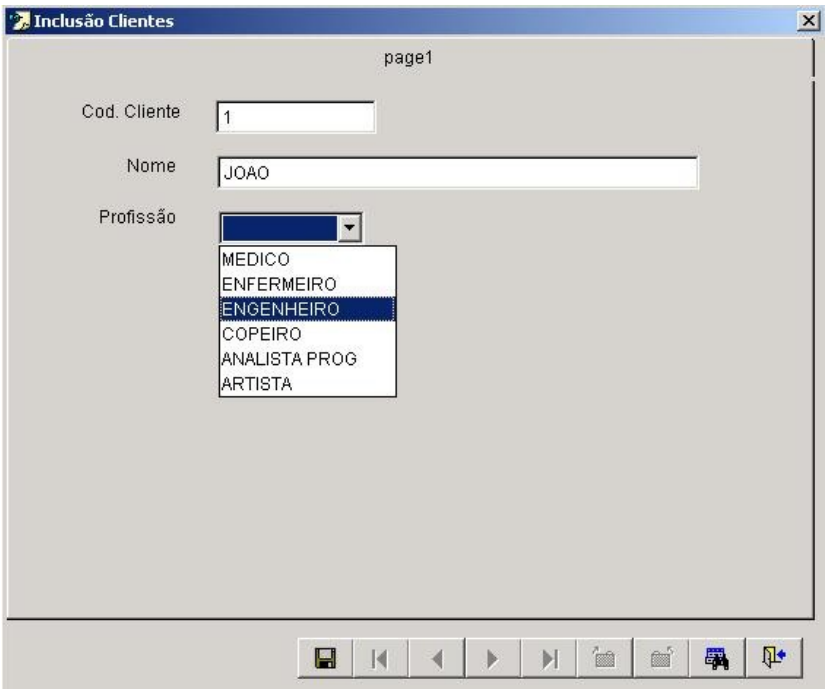


Fig 68 Cadastrando *Clientes* usando o *ComboBox*

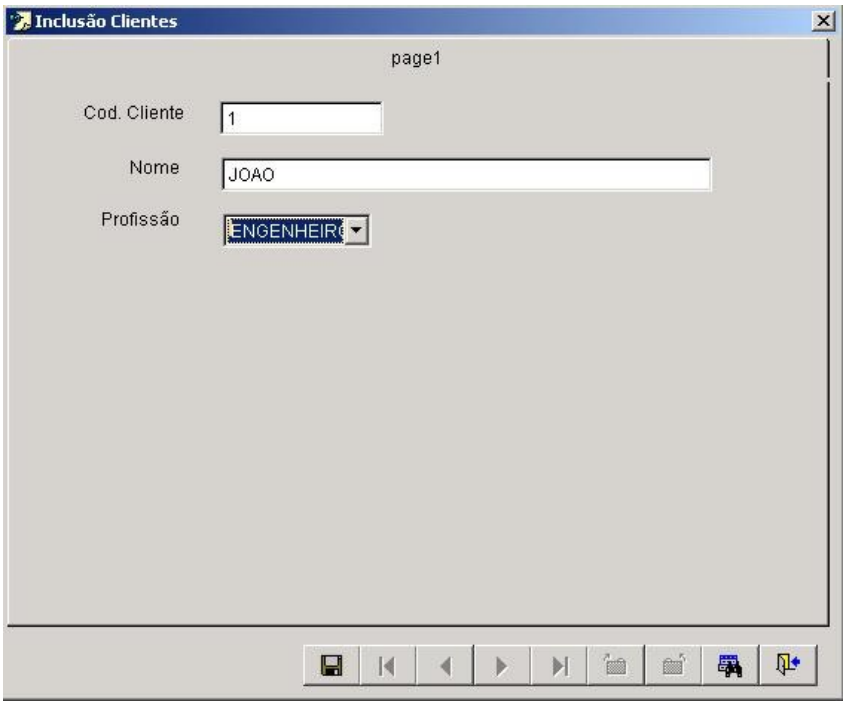


Fig 69 No final da seleção o *ComboBox* fecha novamente.

CheckBox e *OptionGroup*

(a) **(b)**

Fig 70 Visual dos objetos *CheckBox* (a) e *OptionGroup* (b)

O *CheckBox* e o *Option* são controles que permitem disponibilizar opções ao usuário, funcionando como um "flag" que poderá ser ativado ou desativado.

Eles são bastante semelhantes, no entanto, o *Option* sempre deverá ser utilizado em grupo, permitindo oferecer opções ao usuário, para que ele possa selecionar uma entre várias disponíveis. Já o *CheckBox* é uma caixa de verificação individual, que pode conceitualmente ser usada em grupo colocando-a juntas, onde o usuário pode ativar ou desativar quantos *CheckBox* forem necessários.

Seguindo o exemplo de cadastro de clientes, um controle *Option* poderia identificar o sexo, masculino **ou** feminino, já um conjunto de *CheckBox* indicaria por exemplo se o cliente possui casa própria, carro, cartão de crédito, plano de saúde, etc.

A inclusão do objeto *CheckBox* é muito similar ao objeto *TextBox*, basta selecioná-lo e inseri-lo no formulário. Observe que este controle tem certas propriedades que podem parecer iguais, mas que representam coisas bem diferentes:

PROPRIEDADE	DESCRIÇÃO
Nome do Campo	Identifica o campo dentro do formulário e da tabela associada. Número máximo de caracteres para o nome é dez (10).
Título do Campo	Descreve o campo em operações de consultas e relatórios.
Texto	É o rótulo associado ao controle que aparece na tela do formulário.

Quando incluímos um *Option* na tela do formulário, aparecerá uma janela **Assistente de Criação.** Nesta janela definimos o número de opções no grupo e a descrição das opções. Veja na Fig 71. Caso precise fazer alterações no controle depois de inserido, selecione-o e clique com o botão

direito do mouse, aparecerá um menu conforme aparece na Fig 72; clique agora em Editar e aparecerá novamente a janela **Assistente de Criação.**

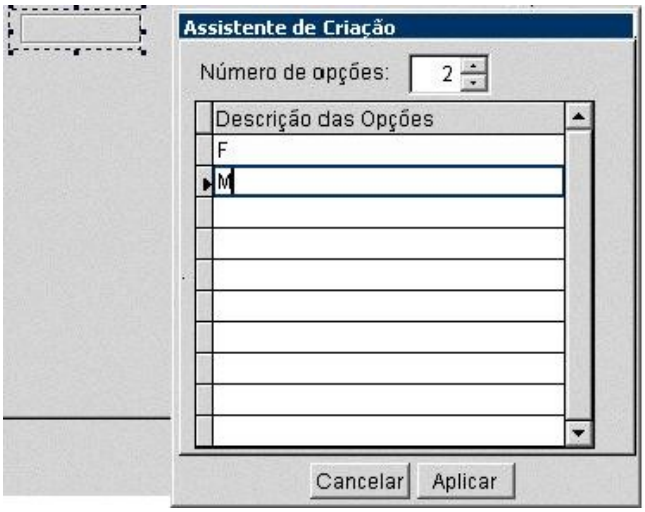


Fig 71 Janela Assistente de Criação do controle *Option*

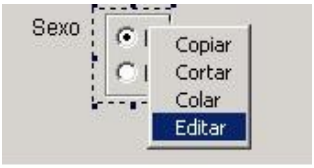


Fig 72 Editando o controle *Option*

Vejamos na Fig 73 como foi complementado nosso formulário com as novas entradas de dados.
.

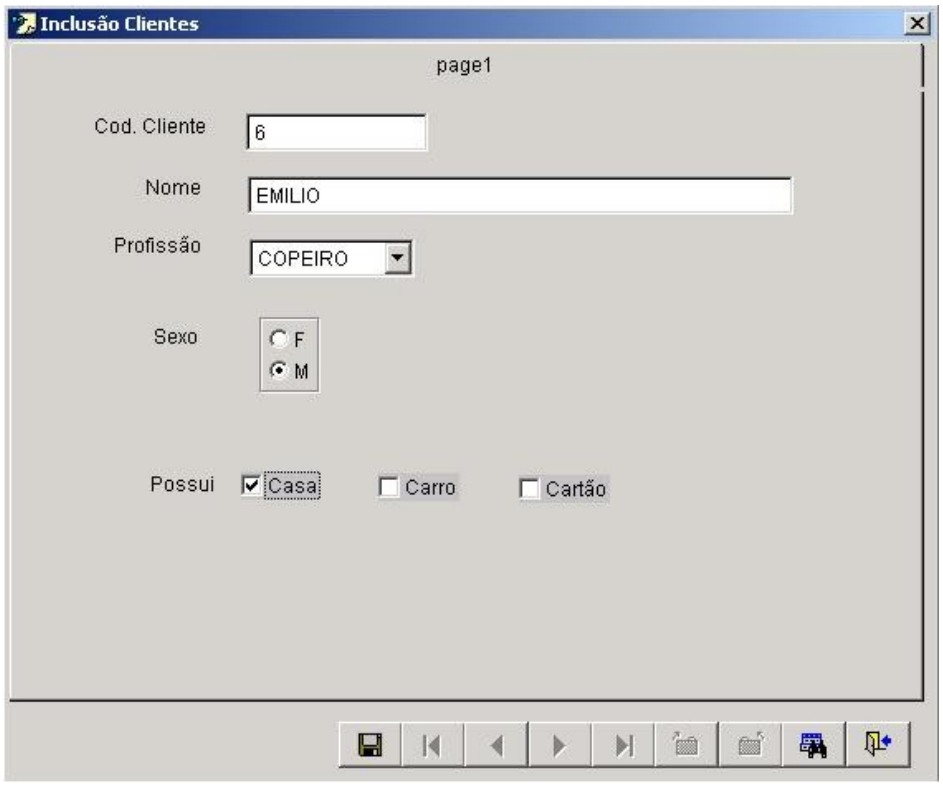


Fig 73 Formulário *Clientes* com *ComboBox, Option* e *Checkbox*

Na Fig 74 mostramos a consulta de vários clientes cadastrados usando este formulário. Note que somente foi criado um campo para o controle *Option* onde: o número **1** indicará sexo feminino e o número **2** o sexo masculino. Por outro lado para cada *CheckBox* foi criado um campo, onde **T**-(True) indica Verdadeiro que significa que foi selecionada e **F** (False) Falso que indica que ficou vazia. Viu que fácil!.

Consulta

Clientes

Codigo	Nome	Codigo Prof	sexo	casa	carro	cartao
1	JOAO	4	2	T	F	T
2	EDUARDO	3	2	T	T	T
3	PEDRO	2	2	T	F	F
4	CELIA	3	1	T	F	T
5	MARIA	1	1	F	F	F

Fig 74 Consulta de clientes cadastrados na entidade *Clientes*.

Spinner

Fig 75 O controle *Spinner.*

O *Spinner* é um controle de rotação, usado para aceitar entrada numérica que pertence a determinado intervalo.

PROPRIEDADE	DESCRIÇÃO
Valor Máximo	Valor numérico máximo que o controle atinge
Valor Mínimo	Valor numérico mínimo que o controle atinge

Imagem

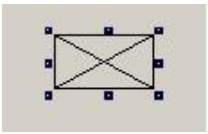


Fig 76 Imagem

PROPRIEDADE	DESCRIÇÃO
Imagem	Caminho da imagem

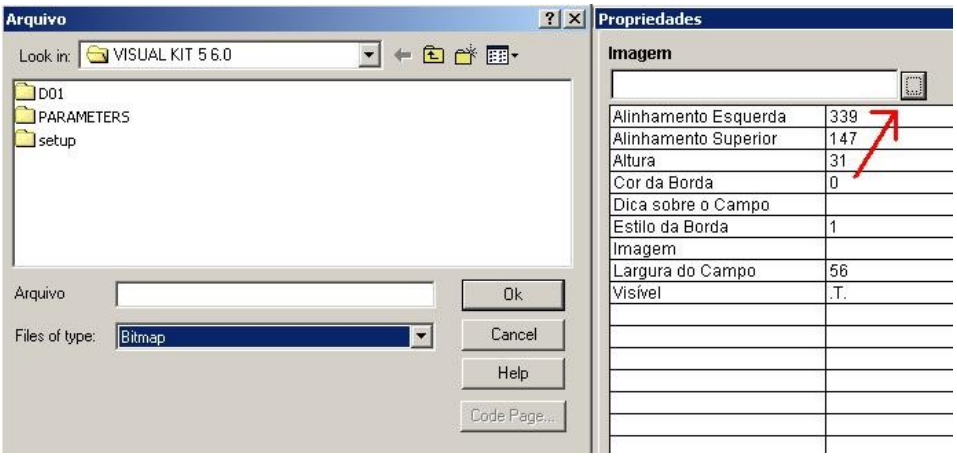


Fig 77

Incluindo funcionalidades nos controles

Quando desenvolvemos sistemas, com certeza, vamos querer que ele realize algumas “cositas más...” além da inclusão, alteração e exclusão de dados. Por exemplo: preenchimento automático do campo, uso de cálculo de vários campos, impedir que um campo fique vazio, aceitar um valor numérico num determinado intervalo, etc.

Para poder colocar funcionalidades deste tipo, basta preencher uma (ou as duas) das interessantes “propriedades” de *Pré-validação* e *Pós-validação* com pequenos códigos que descrevemos neste capítulo.

Quando este código é colocado na *Pré-validação* de um determinado campo, ele é executado quando o campo é focado ou acessado. Os códigos colocados na *Pós-validação* serão executados após a saída do mesmo. Pode ser que determinado campo tenha simultaneamente *Pré e Pós-validação*.

Nota

Foco: Diz-se que um controle num formulário ganha ou perde o foco quando ele se torna ou deixa de ser o objeto atualmente selecionado, seja por ação do usuário ou programaticamente. Esses eventos são temporalmente vizinhos.

Veja na Fig 78 onde são localizadas a *Pré-validação* e *Pós-validação* na janela de propriedades.

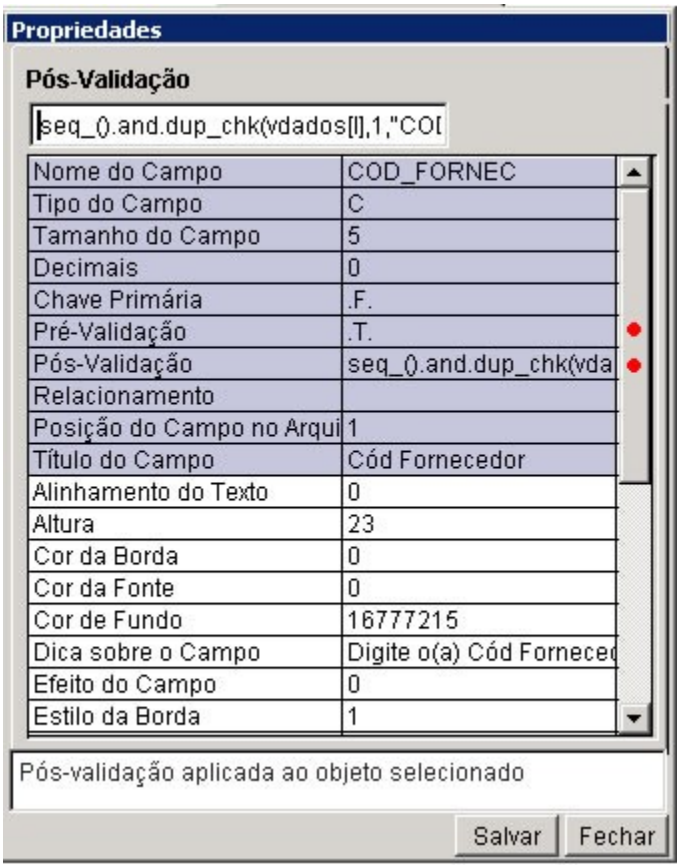


Fig 78 Pré-Validação e Pós-Validação na janela de propriedades.

Algumas das funcionalidades a serem criadas usando as funções pré e/ou pós-validação de um determinado campo precisaram acessar o conteúdo de outros campos do formulário. Como, estando focado em um determinado campo, acessar o conteúdo de outro campo? – Isto será respondido ainda neste capítulo.

Acessando dados pelo vetor *Vdados*

No Visual Kit 5, para poder acessar e manipular os dados dos campos durante a pré ou a pós-validação, é usado um registro chamado de vetor *vdados.*

Nota

Vetor: é uma coleção de dados que compartilha o mesmo nome e possui um número, chamado de índice, o qual permite identificar o elemento que desejamos manipular.

A quantidade de elementos deste vetor coincide sempre com a quantidade de campos que o formulário contém. Na Fig 79 colocamos um exemplo abstrato de um simples formulário com quatro campos para melhor entendimento.

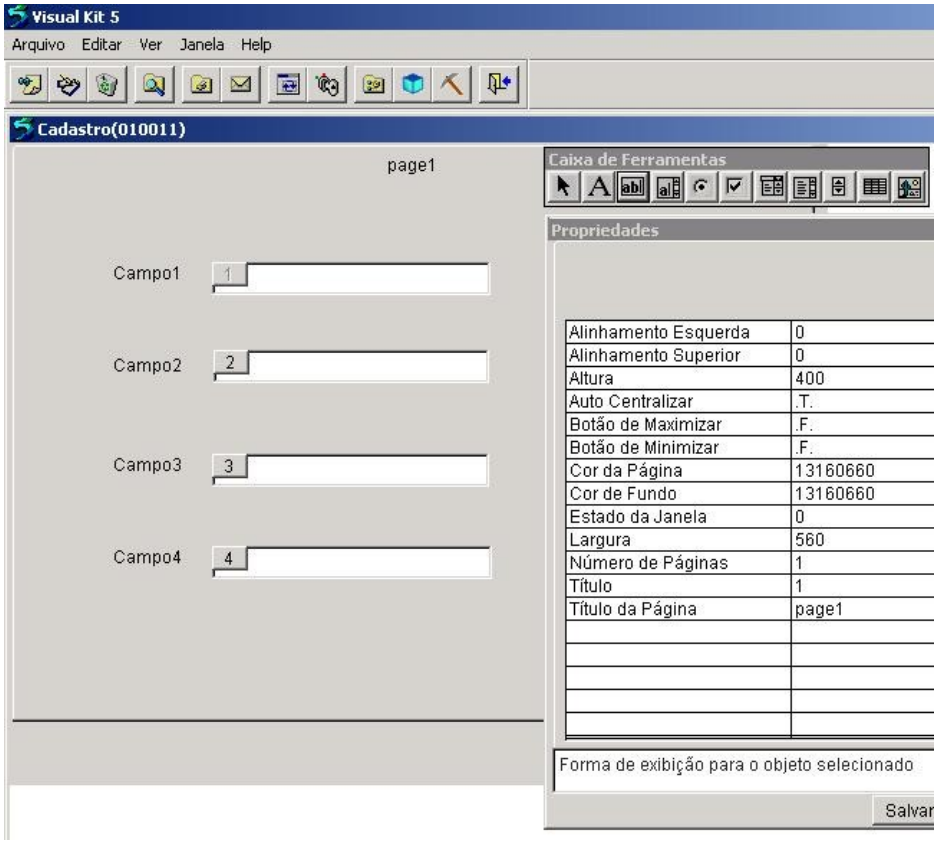


Fig 79 Índices do vetor *vdados*.

Neste caso o vetor *vdados* é formado por quatro (4) elementos. Os elementos do vetor são acessados de duas formas, de maneira direta e de maneira referenciada.

Na primeira, o vetor indicará o conteúdo dos campos através de índices da seguinte forma:

Campo 1	vdados[1]
Campo 2	vdados[2]

Campo 3	vdados[3]
Campo 4	vdados[4]

Na janela de propriedades existe uma propriedade chamada de "*Posição do Campo no Arquivo"* que indicará a posição do campo dentro do vetor *vdados*. Esta posição pode ser modificada mudando a ordem de tabulação, descrita anteriormente.

A outra variante para acessar o valor do campo através do vetor *vdados,* é fazendo referência com o próprio campo focado -dono das propriedades de pré ou pós-validação- e que explicaremos a continuação usando um exemplo.

No exemplo colocaremos as duas maneiras de acessar o vetor *vdados*: usando o índice direto e referenciando pelo campo focado. Neste momento não é necessário entender como é feito o cálculo no exemplo, e sim compreender como os dados devem ser manipulados através da estrutura *vdados*.

Suponha que você tenha uma entidade de vendas, onde existe os seguintes campos: código do cliente, valor do produto, a porcentagem do desconto cedido e logo após um campo onde entrará o resultado do cálculo total com o desconto. Para calcular o valor do campo *Total* de maneira automática inserimos na propriedade *Pré-Validação* do próprio campo o seguinte código:

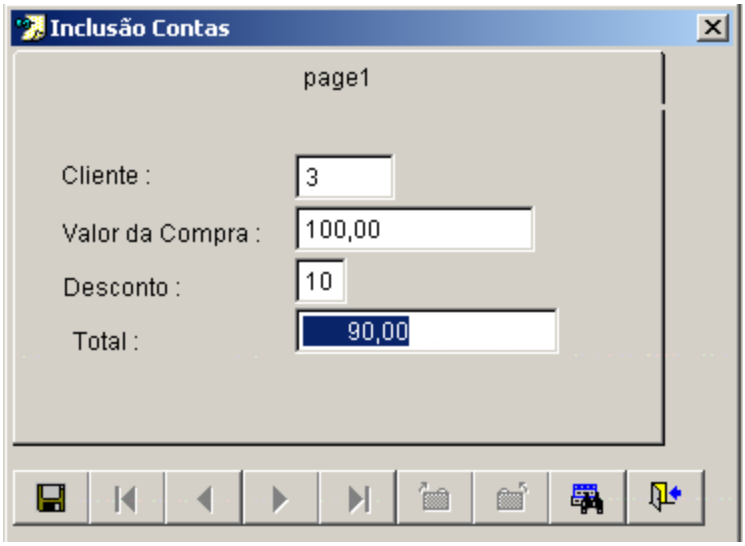


Fig 80 Exemplo usando o vetor vdados

Usando o índice direto:

```
AltVal(vdados, 4, (vdados[2] - (vdados[2] * vdados[3])/100))
```

onde, *vdados[2]* é o Valor da Compra e *vdados[3]* é o Desconto.

Vejamos agora usando o índice referenciado ao campo focado (neste caso o campo *Total*). O acesso é feito da seguinte forma:

Código do cliente	Vdados(i-3)
Valor da compra	Vdados(i-2)
Desconto	Vdados(i-1)
Total	Vdados(I) Campo focado na pré-validação

A propriedade Pré-Validação tem agora o seguinte código:

```
AltVal(Vdados, I, (Vdados[i-2] - (Vdados[i-2] * vdados[i-1])/100))
```

Partindo do princípio que o cursor acabou de focar ou entrar no campo *Total* (Vdados[i]), o cálculo será realizado a partir do campo *Valor da Compra* (Vdados[i-2]) porque está a duas posições acima do campo com o foco, e do campo *Desconto* – que é o campo anterior (Vdados[i–1]).

As duas formas, são análogas e dão o mesmo resultado. A diferença radical é que, se por acaso, é decidido incluir mais um campo seguido ao código do cliente, com por exemplo o *Nome do Cliente*, será preciso modificar o código feito usando o índice direto. Já o código feito usando o índice referenciado não precisará ser mudado, se é mantida a ordem dos campos acessados na pré-validação.

Áreas de relacionamento - acessando conteúdos de outras entidades.

Na seção anterior vimos como é fácil acessar o conteúdo dos campos dentro de um mesmo formulário ou entidade. Vejamos agora a maneira de acessar campos de outras entidades relacionadas.

O acesso a campos de outras entidades e feito através das *Áreas de Relacionamento*. Quando uma entidade do sistema é aberta, automaticamente são disponibilizadas diferentes áreas de memórias do computador. É aberta uma área para a própria entidade e tantas quantos relacionamentos tenha com outras entidades.

Áreas de relacionamento são diferentes áreas de memórias do computador que são disponibilizadas para cada relacionamento do formulário que o usuário esteja usando no dado momento. Estas áreas são identificadas com números inteiros, ex. *1*; *2*; *3*; etc. e o número vai a depender da ordem no formulário dos campos com relacionamento. Por exemplo:

a) Área *1*: área aberta para a própria entidade.
b) Área 2: área para a primeira entidade relacionada pelo campo, segundo a "*Posição do Campo no Arquivo"* ou ordem de tabulação.
c) Área *3*: área para a segunda entidade relacionada. E assim por diante.

Vejamos no exemplo: digamos que você criou uma sistema com três entidades: *Clientes*, *Produtos* e *Pedidos*. E que a entidade *Pedidos* como é de supor está relacionada com *Clientes* e com *Produtos*. Se dentro do formulário *Pedidos* o campo de relacionamento com *Clientes* tem o valor da propriedade "*Posição do Campo no Arquivo"* menor que o do campo de relacionamento com a entidade *Produtos*, então quando acessamos o formulário *Pedidos,* as áreas abertas estarão da seguinte maneira.

a) Área *1*: área aberta para *Pedidos*.
b) Área 2: área aberta para *Clientes*.
c) Área *3*: área aberta para *Produtos*.

Agora, para acessar um campo específico da entidade relacionada é utilizada a seguinte notificação:

letra_area->nome_campo.

onde, a *letra_area* é a letra do abecedário (Inglês) correspondente ao número da área de relacionamento, e o *nome_campo* é o nome do campo específico que queremos acessar.

Seguindo o exemplo anterior, se por exemplo, dentro do formulário *Pedidos* usamos alguma *Pré* ou *Pós-validação* que precisa manipular o campo *Nome* da entidade *Cliente* e o campo *Tipos* da entidade *Produtos*, estes campos serão identificados da seguinte maneira:

Campo *Nome* de *Cliente*: **b->Nome**
Campo *Tipo* de *Produtos*: **c->Tipos**

Vejamos agora como estes conceitos aqui abordados são usados nas funções de validação.

Funções de validações (pré e pós)

Nesta seção colocaremos as validações que podem ser inseridas na Pré e Pós Validação na janela de propriedades.

Exibe caixa de mensagem: Função *ferrmsg()*

Exibe caixa com mensagem na parte superior direita do monitor.

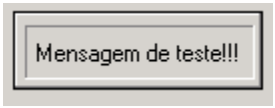


Fig 81 Caixa da mensagem

Sintaxe:

ferrmsg("*texto da mensagem*", *segundos***)**

em que:

- ***texto da mensagem:*** expressão em caracter (*string*) que será apresentada como mensagem na caixa exibida pela função.
- ***segundos:*** especifica a quantidade de segundos que a mensagem vai permanecer.

Observações:

- O texto da mensagem. deve estar entre aspas duplas (“...”).
- Caso você queira colocar o conteúdo de um campo numérico na mensagem, deverá utilizar a função ***str()*** que transforma o valor numérico em caracter. Veja nos exemplos.
- Caso não seja colocado o valor em segundos, o programa aguardará até ser pressionada uma tecla.
- Esta função é ideal para colocar mensagem e erro para o usuário.

Exemplos

ferrmsg(“Data invalida”) : exibirá a mensagem e aguardará até o operador pressionar alguma tecla

ferrmsg(“Hora invalida”,2): exibirá a mensagem por 2 segundos

ferrmsg(“você digitou”+ **str**(vdados(I))): Fará uma concatenação da expressão em caractere com o campo numérico usando o signo de soma.

Conteúdo automático - Função *altval()*

Esta função insere ou altera automaticamente o conteúdo de um campo acessando através do vetor *vdados*.

Sintaxe:

altval*(vdados,I,Novo valor)*

em que:

- **vdados**: o vetor de entrada de dados do sistema.
- **I**: posição ou índice atual do campo;
- **Novo Valor:** valor que será atribuído ao campo.

Vejamos em exemplos concretos como funciona esta função:

Retorna cálculos matemáticos

*altval(vdados, I, ((vdados[I-1] * 8) / 100))*

Neste caso, estamos atualizando o campo atual com 8% do valor do campo anterior. Este campo e o campo anterior devem ser de tipo NUMÉRICO.

*altval(vdados, I, (vdados[I - 1] * vdados[I - 2]))*

Neste caso, atualiza o campo atual com a multiplicação dos campos anteriores. Este campo e os campos anteriores devem ser de tipo NUMÉRICO.

Somar dias a um campo data

altval(vdados, I, vdados[I] + 15)

Neste caso, soma 15 dias à data digitada. Este campo deverá ser um campo do tipo DATA.

Retorna o dia de uma data

altval(vdados, I, day(date()))

Neste caso, pega o dia da data do sistema. Este campo deverá ser um campo do tipo NUMÉRICO.

Retorna o mês de uma data

altval(vdados, I, month(date()))

Pega o mês da data do sistema. Este campo deverá ser um campo do tipo NUMÉRICO.

Retorna o ano de uma data

altval(vdados, I, year(date()))

Neste caso estamos pegando o ano da data do sistema. Este campo deverá ser um campo do tipo NUMÉRICO.

Retorna hora do sistema

altval(vdados, I, time())

Neste caso, coloca a hora do computador no campo atual. Este campo deverá ser um campo do tipo CARACTER com 5 posições.

Retorna conteúdo de entidade relacionada.
altval(vdados, I, B->NOME)
Coloca o campo NOME de uma entidade relacionada no campo atual. Este campo deverá ser um campo de tipo e tamanho idênticos ao campo da entidade relacionada.

Coloca valor *default* - Função *rep_()*

Preenche o campo automaticamente com o valor passado como parâmetro. Agilizando a digitação do usuário do sistema. Caso necessário, este valor poderá ser modificado pelo ousuário.

Sintaxe:
rep_(*exprCN*)
em que:
exprCN- é o conteúdo a ser colocado como *default*.

Exemplos:
rep_("RIO DE JANEIRO"): Coloca “RIO DE JANEIRO” no campo. Ideal para casos como: formulário de cadastro de clientes onde a maioria more em uma cidade em específica. rep_(021): Repete 021 em campo numérico. Ex.: código de área.
rep_(date()) Repete a data de hoje sempre que teclar ENTER.

Cria código seqüencial automático - Função *seq_()*

Cria um código seqüencial.

Sintaxe:
seq_()
em que:

Esta função cria um número seqüencial tanto em campos caracteres como em campos numéricos. Ao teclar “ENTER” no campo, ela pega o último valor digitado no arquivo e acrescenta um, senão aceita o valor digitado.
Ideal para ser utilizada em campos que armazenam algum código tipo Código do Cliente.

Impede campo vazio-Função *!empty()*

Verifica se um campo é ou não vazio. Retorna TRUE (.T.) se o campo for vazio, independente do seu tipo (caractere, numérico, data...).

Sintaxe:
!empty(*exprC*)
em que:
exprC: referencia ao campo pelo vetor *vdados*

Exemplo
!empty (vdados[I]) ; Impedirá que o campo atual fique vazio, ideal para campos importantes e usuários não atenciosos ...

Limita o conteúdo do campo

Em determinadas ocasiões queremos colocar determinadas condições ou limites, por exemplo: o tipo de caracter que poderá ser armazenado ou um valor numérico em determinado intervalo. Vejamos em exemplos como pode ser feito.

Aceita S (Sim) ou N(Não)

(vdados[I] $ "SN")
A função **$** verifica se uma seqüência de caracteres está contida na expressão. Retorna TRUE (.T.) se a seqüência for encontrada.

Aceita valor num intervalo

O intervalo é criado usando os operadores lógicos AND e OR que significam respectivamente E e OU.

(vdados[I] >= 1 .and. vdados[I] <= 10)
Aceita valor no campo atual no intervalo maior ou igual a 1 e menor ou igual a 10. Ou seja, vdados[I] deve ser ao mesmo tempo maior ou igual que 1 **e** menor ou igual a 10.

(vdados[I] > 100 .or. vdados[I] < 10)
Aceita valor no campo atual se o valor for maior que 100 **ou** menor que 10.

Aceita um período de datas

(vdados[I] >=(date()-15) .and. vdados[I] <= date())
Aceitas datas entre dos últimos 15 dias

Não permite conteúdo duplo - Função *dup_chk()*

Faz pesquisa indexada para verificar se o registro que está sendo digitado se existe ou não no arquivo. Se o registro já existir, exibe mensagem de erro e não permite a inclusão.

Sintaxe:

dup_chk(vdados[I], Área,"mensagem",tipo,I,impre[,order])

em que:

- **vdados[I]**: é o vetor vdados indexando o campo em que contém o valor que será pesquisado;
- **Área**: área de relacionamento onde o arquivo que será pesquisado foi aberto;
- **"mensagem":** mensagem de erro (entre aspas) que será exibida.
- **tipo**: variável ou valor lógico. *Verdadeiro* se a operação a ser executada for inclusão. *Falso* se for outra operação. A variável tipo recebe um valor lógico e é uma palavra reservada do sistema. É proibido criar um campo chamado "tipo".
- **I**: indexador, que indica em qual elemento do vetor vdados o sistema está efetuando a entrada de dados. Este argumento é representado pela letra "I" maiúscula.
- **Impre**: variável lógica, que recebe um valor: *Verdadeiro* se a operação a ser executada for de impressão, *Falso* se for outra operação. A variável **impre** recebe é uma palavra reservada do sistema. É proibido criar um campo chamado "impre".
- **order**: argumento opcional e representa a ordem do indexador (set order).

Observação:
Esta função retorna um valor lógico: *verdadeiro* se o valor pesquisado foi localizado, *falso* se não.

Exibe conteúdo na tela – Função *prt ()*

Imprime na tela do formulário usando um controle *Label* (Rotulo), o conteúdo de um campo da entidade atual ou relacionada.

Sintaxe:

ptr(linha,coluna, conteúdo,fonte,tamanho,label,page)

em que:

- **linha**: Posição superior do objeto *Label*
- **coluna**: Posição a esquerda do objeto *Label*
- **conteúdo**: Pode ser o próprio conteúdo a ser exibido, ou algum campo referenciado **pelo** vetor vdados ou campo relacionado pela área
- **fonte**: Tipo de fonte a ser usada. "Arial", "Courier", etc.

- **tamanho**: Tamanho da fonte. Numero inteiro
- **label**: nome do *Label* usado para a impressão. Ex: Label005
- **page**: Como o formulário pode possuir inúmeras páginas. Este número indica a página onde está o controle *Label*

Exemplo:
prt(162,68,vdados[I],"Arial",8,"label003",1)
Imprime na janela do formulário na posição (162,68) o conteúdo do campo atual.

prt(162,68,b->nome,"Arial",8,"label003",1)
Exibe na janela do formulário, o conteúdo de um campo de uma entidade relacionada aberta na área 2. Ideal em casos deste tipo: Uma entidade de *Pedidos*, relacionada a uma entidade de *Clientes* pelo código do cliente. Digitando o código do cliente em *Pedidos*, imprime o nome do cliente na tela.

Salta campo - Função *slt()*

Salta campo slt() . A função slt(fieldN,scrN) pula para o campo desejado.

Sintaxe:
slt(fieldN,scrN)
em que:
- **FieldN** Campo para onde irá o cursor.
- **ScrN** Tela onde está o campo

Exemplo
slt(15,1) : O cursor pula para o campo 15 do formulário 1

Pesquisa se chave existe – Função *dbseek( )*

Faz uma pesquisa indexada e retorna TRUE se encontrou a chave e FALSE se não encontrou a chave. Esta função pode ser utilizada em qualquer campo que possua um indexador ativo.

Sintaxe:
dbseek(Expr, [ExprN1], [ExprN2])
em que:
- **Expr**: Campo referenciado pelo vetor vedados e que será pesquisado.
- **ExprN1**: Área de abertura para pesquisas em entidades relacionadas.
- **ExprN2**: ordem do indexador ativo.

Exemplos:
dbseek(vdados[I],2): Pesquisa na área 2 se vdados[I] existe como chave.
dbseek(vdados[I],1,2) Pesquisa na área 1 order 2 se vdados[I] existe como chave.

Busca última ocorrência da chave - Função *bprx( )*

Busca a última ocorrência de uma chave na área e order especificados.

Sintaxe:
bprx(ExprC1, ExprN1, ExprN2, ExprC2)
em que:
- ExprC1: Chave para pesquisa.
- ExprN1: Área de abertura.
- ExprN2: Order (set order).
- ExprC2: Campos que serão comparados com a chave.

Enquanto a chave for igual aos campos lê arquivo, quando encontrar alguma diferença retorna 1 registro e volta. Se não for encontrado nenhum registro que satisfaça a condição retorna EOF() TRUE.

Exemplo
bprx(vdados[1]+vdados[2], 1,1 "CONTA+BANCO")

Altera conteúdo de outras entidades - Função *chg_data ()*

Altera dados em entidades relacionadas com a entidade ou formulário atual

Sintaxe:
chg_data(expN1 , expC1 , expL, expC2 , fieldN, expCN3)
em que:
- **expN1**: Área de abertura do arquivo que será atualizado;
- **expC1**: Chave para pesquisa da primeira ocorrência no arquivo que será atualizado;
- **expL:** Se argumento verdadeiro (.T.) a atualização será efetuada no primeiro registro encontrado. Se argumento (.F.) a atualização será efetuada sob uma determinada condição referenciada no próximo parâmetro;
- **expC2**: Condição para atualização de um grupo de registros;
- **fieldN**: Número do campo que será atualizado.

- **expC3**: Expressão caracter ou numérica que será atribuída ao campo destino.

É possível alterar dados em um único registro ou num grupo de registros (while).

Exemplo
chg_data(2 , vdados[1] , .t. , “” , 7, B->QUANTIDADE+vdados[I])
Neste exemplo, estaremos alterando o campo número “7”, da entidade aberta área “2”, relacionada pela chave “vdados[1]”, e somando ao campo “QUANTIDADE” do arquivo na área “2” o valor digitado em “vdados[I] da entidade atual”.

Calcula o módulo 11 - Função *mod11( )*

Calcula o módulo 11 de um campo e grava o dígito verificador (DV) no próximo campo

Sintaxe:
mod11(*exprC, [ExprC]*)

A função mod11() calcula o módulo 11 do campo passado como parâmetro e coloca o valor calculado (dígito verificador de 1 posição) no campo atual. O segundo parâmetro é opcional, deve ser uma letra. Será o dígito verificador em caso de resto igual a 10.

Exemplos:
mod11(vdados[I-1]) Calcula o módulo 11 do campo anterior e coloca o resultado no campo atual.
mod11(vdados[I-1], “P”) Calcula o módulo 11 do campo anterior e coloca o resultado no campo atual. Se resto igual a 10: DV igual a P.

Verifica hora - Função *validtime( )*

Verifica se o conteúdo da variável passada é uma hora válida

Sintaxe:
validtime(vdados[I])
em que:

Exemplo: 14:00.
O formato deste campo deve ser: “@R 99:99”.
Esta função retorna um valor lógico: *verdadeiro* se a hora digitada for válida, *falso* se não for.

Altera variáveis do sistema - Função *altvar( )*

Altera valor de variável de programa do Visual Kit 5.

Sintaxe:

altvar*(t_n, Novo valor)*

em que:

- **t_n**: variável que será alterada.
- **Novo valor**: valor que será atribuído ao campo.

Variáveis são pequenos espaços de memória que armazenam valores para uso geral sendo que valor armazenado poderá ser modificado durante a execução do programa. O Visual Kit 5 dispõe cinco (5) variáveis numéricas (só armazenam números) para serem utilizadas em cálculos gerais como totalizações nos sistemas criados. Estas variáveis são representadas como: **t_1**; **t_2**; **t_3**; **t_4** e **t_5** e são todas inicializadas com valor zero.

Elas servem para armazenar valores entre os registros das tabelas, enquanto a inclusão estiver rodando, ou seja, caso o usuário feche a inclusão e a abra novamente, as variáveis perderão os seus valores armazenados.

Alterar tamanho campo memo -Função *altmemo ()*

Altera o tamanho da linha do campo tipo Memo na impressão de lay-out.

Sintaxe:

altmemo(ExprN)

em que:

- **ExprN**: novo número de colunas por linha no campo tipo Memo.

O valor *default* de colunas por linha no campo tipo Memo é de 78, você pode alterar este valor utilizando esta função. Não esqueça de retornar ao valor original com altmemo(78).

Exemplo:

altmemo(90) Altera o valor da linha de 78 para 90.

Seleciona área de relacionamento - Função *dbselect( )*

Seleciona uma área de trabalho.

Sintaxe:
dbselect(ExprN)
em que:
- **ExprN**: Número inteiro que indica a área de relacionamento.

Não esqueça que sempre que for selecionada uma área de trabalho você deve retornar a área atual com dbselect(1).

Exemplos:
dbselect(2) seleciona área 2.
dbselect(3) seleciona área 3.

Retorna ao registro atual - Função *goatu( )*

Posiciona o ponteiro do arquivo no registro que está sendo incluído ou alterado.
Esta função deve ser utilizada sempre que você alterar através da pesquisa o ponteiro do registro atual para outro registro no arquivo corrente. Se você alterar o ponteiro do arquivo e não retornar, poderá ocorrer um erro no momento da gravação do registro.

Vai para o início do arquivo - Função *dbgotop( )*

Posiciona o ponteiro do arquivo no primeiro registro.

Sintaxe:
dbgotop()

Vai para o final do arquivo - Função *dbgobottom( )*

Posiciona o ponteiro do arquivo no último registro.

Sintaxe:
dbgobottom( )

Movimenta o ponteiro do arquivo - Função *dbskip( )*

Movimenta o ponteiro do arquivo.

Sintaxe:

dbskip([ExprN])

em que:

- ExprN: é um número inteiro. Se positivo, movimenta o ponteiro para frente, se negativo, movimenta o ponteiro para trás. O valor indica os números de registros.

Exemplos

dbskip(2) movimenta o ponteiro dois registros para frente.

dbskip(- 3) movimenta o ponteiro três registros para trás.

Grava dados em campo - Função *fieldput( )*

Grava dados num determinado campo.

Sintaxe:

fieldput(FieldN, Valor)

em que:

- FieldN: número inteiro que indica o campo.
- Valor: Conteúdo a ser gravado.

Exemplo

fieldput(2, "EBENDINGER"): grava no campo número 2 do formulário atual o nome EBENDINGER.

fieldput(3, "BRASIL"): grava no campo número 3 do arquivo atual o nome BRASIL.

Exibe um controle objeto-imagem no formulário – Função *prt_image( )*

Exibe qualquer dado na tela. Este dado pode ser um campo do arquivo atual, campos de arquivos relacionados, valores numéricos, caracter, data e qualquer informação que possa ser exibida no monitor

Sintaxe:

prt_image(ExprN1, ExprC1, ExprN2)

em que:

- **ExprN1**: Número que representa a posição no arquivo do campo que contém o caminho da Imagem. ex: c:\widows\imagem.bmp
- **ExprC1**: Nome do controle imagem criado pelo Visual Kit 5. ex: "image001"
- **ExprN2**: Número da página onde será exibida a imagem.

Exemplo:
prt_image(2,"image001",1)

Observação: A imagem somente será exibida ao passar pelo campo que contém esta função. Não colocar a função Prt_image() antes do campo que contém o caminho da Imagem.

Criando validações compostas

Usando o operador AND

Você pode digitar uma ou mais funções no campo “Pré ou Pós validação de Campo”, usando o operador AND para separar as funções.

Exemplo: seq_() and ferrmsg(“Ao final salve tudo”)

Usando o operador *iif()*

O operador *iif()*, também conhecido como operador condicional poderá determinar qual função executar avaliando primeiramente uma expressão lógica.

Sintaxe:
iif(ExprL, validação1, validação2)
em que:

- **ExprL**: Expressão lógica que deve retornar Verdadeiro ou Falso.
- **validação1**: Validação a executar caso *ExprL* seja verdadeira
- **validação2**: Validação a executar caso *ExprL* seja falsa.

Observações:

- A expressão lógica *ExprL* pode ser composta de várias sub-expresões usando os operadores AND (E) e OR (OU)
- Uma própria validação , ex. **validação1** pode conter outro operador *iif()*

Exemplos:

a) **iif(vdados[4]>1000,altval(vdados,4,vdados[4]* 5),altval(vdados,4,vdados[4]* 25))**

Se o conteúdo do quarto campo for maior que 1000, multiplica o valor por 4, do contrário multiplica por cinco.

b) **iff(vdados[3]>=1 and vdados[3]<10, ataval(vdados,3,vdados[3]*10), ferrmsg(“Número fora do intervalo!!”))**

Se o conteúdo do terceiro campo é maior ou igual que 1 e menor que 10, então multiplicá-lo por 10, caso contrário exibe caixa de mensagem.

c) **iff(vdados[3]>=1 and vdados[3]<10, iif(vdados[4]>1000,altval(vdados,4,vdados[4]* 5),altval(vdados,4,vdados[4]* 25)),ferrmsg(“Número fora do intervalo!!”))**

Se o conteúdo do terceiro campo é maior ou igual que 1 e menor que 10, então fazemos outra avaliação *iif*: (Se o conteúdo do quarto campo for maior que 1000, multiplica o valor por 4, caso contrário multiplica por cinco); do contrário exibe caixa de mensagem.

VII. A *Designer-Pro*: Novos Recursos

Nesta seção abordaremos os recursos da *Designer–Pro,* que consiste em novos controles na caixa de ferramentas, assim como novos assistentes de criação e os editores de *Expresões*, *Macros* e *Código*.

Para acessar a *Designer–Pro* sempre que você definir uma nova entidade selecione **Formulário Externo** (veja a Fig 42). Será aberta uma interface muito similar à Designer Básica mais com muitos mais recursos.

Como você deve ter notado, alguns controles estão disponíveis em ambas *designers*, caso não exista mudança na nova designer, não estaremos comentando o controle.

Repetiremos para vários controles exatamente os mesmos procedimentos que na *Designer-Básica*. Note a diferença usando os assistentes de criação.

Principais controles da *Designer-Pro*

Novo *ListBox*

Fig 82 Visual do *ListBox*

O controle cria uma caixa de listagem, onde exibe uma lista de itens para que o usuário possa selecioná-los. Você deverá utilizar o *ListBox* sempre que tiver de fornecer várias opções ao usuário, como por exemplo, uma lista de profissões dentro de um cadastro de clientes. A lista pode ser

rolada se tiver um número de itens superior ao que pode ser exibido de uma só vez.

PROPRIEDADE	DESCRIÇÃO
Relacionamento	Identifica a entidade a que estará relacionado o controle. Na entrada de dados será mostrada os nomes das tabelas segundo o padrão Visual Kit 5.
Dica sobre o campo	Insira aqui o texto que será apresentado em uma caixa de texto amarelada (conhecida como *ToolTip*), quando o ponteiro do mouse ficar sobre o controle por alguns segundos

O controle *ListBox* deverá obrigatoriamente estar relacionado a uma segunda entidade que gerenciará e armazenará a lista de itens. Exemplo de uma entidade com dois campos: o campo chave primária da lista - por exemplo o código da profissão- , mais um outro campo, - por exemplo, nome da profissão.

Você poderá escolher – através de um Assistente de Criação- os campos que apareceram na lista do controle *ListBox*. Este controle sempre gravará na entidade atual o campo chave da entidade relacionada, inclusive quando esta não for selecionada para ser exibida no controle. Para maior compressão implementaremos o seguinte exemplo.

Exemplo de sistema com entidades relacionadas.

A proposta –lembre-se sempre de fazer a proposta antes de começar a desenvolver qualquer sistema: Vamos criar um sistema composto por um módulo e duas entidades. O módulo *Cadastro* com as entidades *Clientes* e *Profissão*. O objetivo da entidade *Clientes* será gerenciar diferentes informações referentes a clientes, por exemplo: nome completo, profissão, sexo, etc. Como exigência o conteúdo do campo profissão deve constar numa lista pré-estabelecida de profissões. O objetivo da entidade *Profissão* então será precisamente gerenciar esta lista.

- Crie um módulo e as duas entidades da mesma maneira explicada no capítulo IV

Fig 83 Criamos o Módulo *Cadastro* com as Entidades *Clientes* e *Profissão*

Desenhemos primeiramente o formulário da entidade *Profissão*.

- Insira dois controles *TextBox* com seus respectivos rótulos (*Label*) identificativos.
- O primeiro campo, será o código da profissão e é campo chave primária. Mude a propriedade **Nome do Campo** para *CodProf* por exemplo e o **Título do Campo** para Código da Profissão.

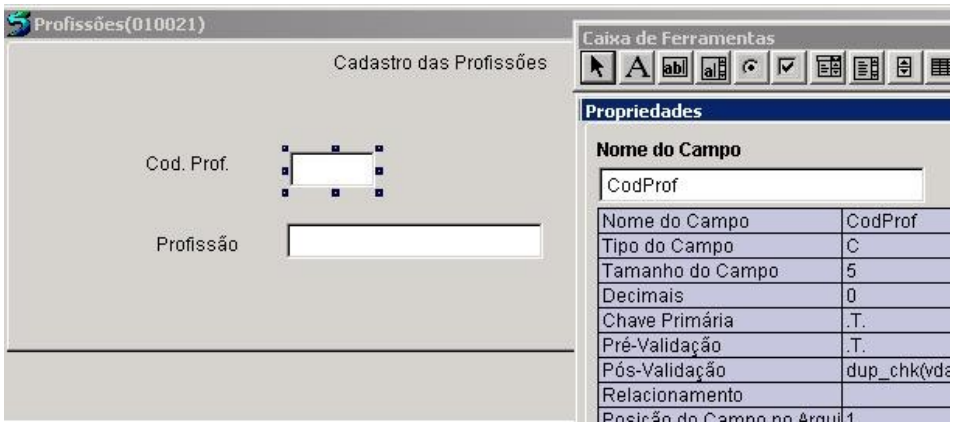


Fig 84 Formulário da entidade Profissões

Atenção: É importante mudar a propriedade **Nome do Campo** dos campos chaves. Se deixar os valores *default*, ao ser relacionado o campo em outra entidade, pode ser que coincida com algum outro nome *default* dessa entidade, aparecendo dois campos com o mesmo nome. Isto causará um erro na execução. Recomendamos colocar nomes para cada campo que o identifiquem segundo o tipo de informação que vai gerenciar.

- Desenhemos agora o formulário *Clientes*. Coloquemos os controles *TextBox* para os campos Cód. do Cliente e Nome.
- Insira um controle *ListBox*. Este campo estará relacionado com a entidade *Profissão,* portanto, é preciso definir a propriedade **Relacionamento**

Para inserir a propriedade de relacionamento:

- Dê um clique sobre a propriedade **Relacionamento**. Aparecerá uma entrada de seleção com as tabelas do sistema (neste caso a entidade *Profissão* **01002**- segundo padrão Visual Kit 5, veja no Apêndice C).

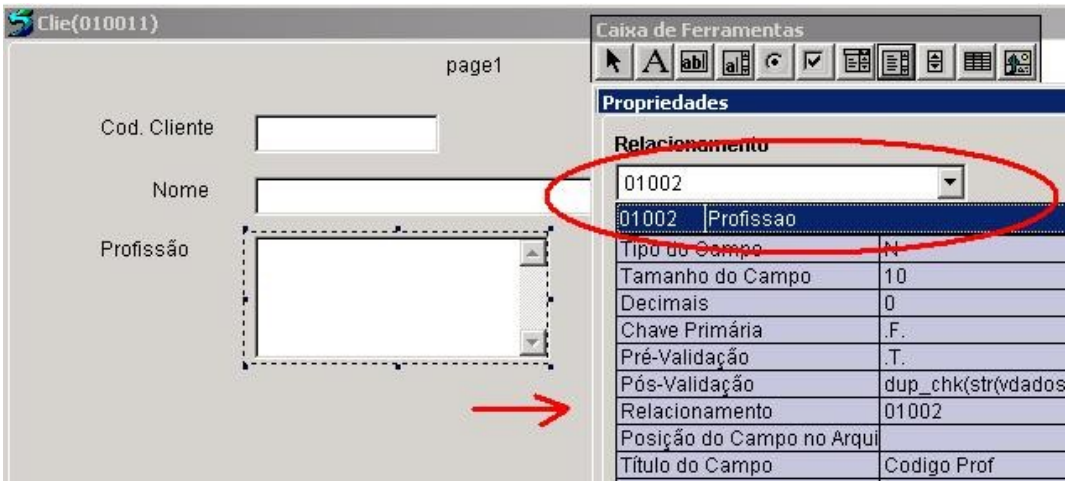


Fig 85 Formulário da Entidade Clientes

Atenção: A novidade: o Assistente de Criação

Selecionada a entidade relacionada, aparecerá uma janela **Assistente de Criação,** onde você complementará o desenho deste controle**.** Esta janela está dividida em duas páginas: **Campos** e **layout** (veja na Fig 86).

Na página **Campos** você seleciona os campos a serem incluídos no controle. Ela dispõe de duas listas lado a lado. Na lista a esquerda (**Campos Possíveis)** aparecem todos os campos disponíveis, e na lista da direita (**Campos Selecionados)** mostra todos os campos selecionados para exibição no controle. Você usa os botões no meio das duas listas para passar campos de uma lista para outra:

- Faça um clique sobre algum campo da lista **Campos Possíveis**
- Clique no botão que indica a direção [>]. O campo agora aparece na lista da esquerda.

Desta maneira você poderá escolher vários campos. Veja nas figuras Fig 86 e Fig 87.

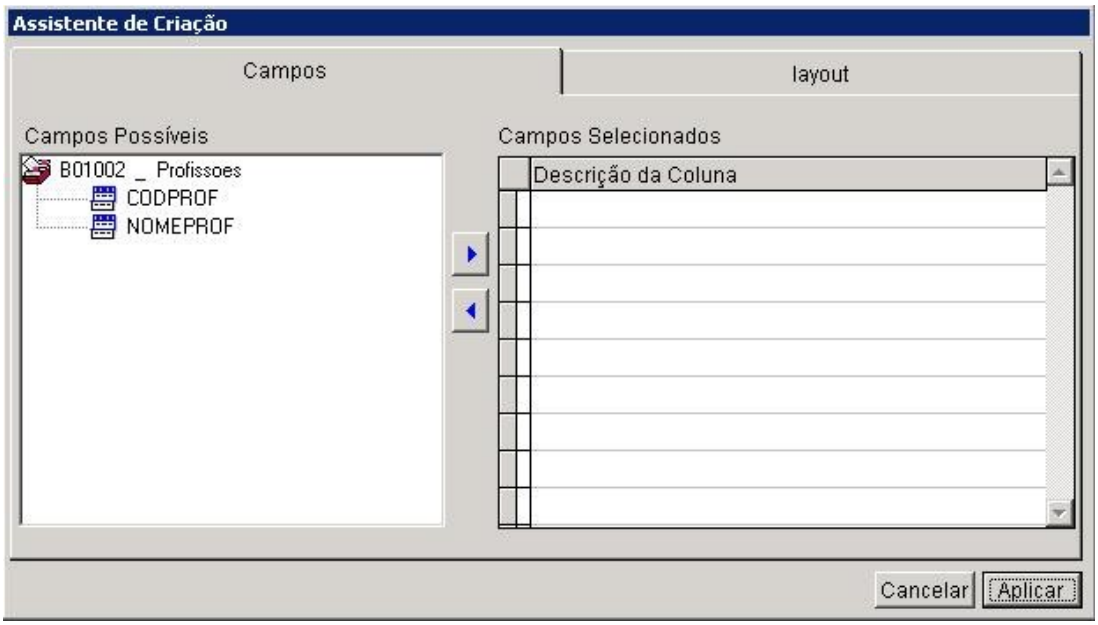


Fig 86 Assistente de Criação.

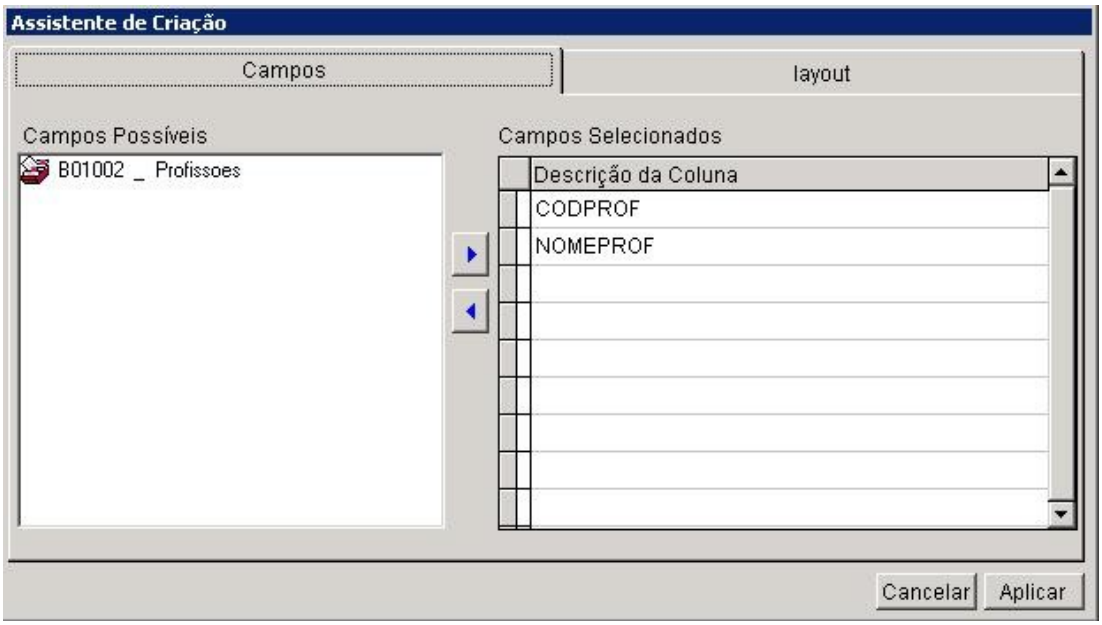


Fig 87 Campos selecionados para ser apresentados.

Na segunda página (**layout**) você seleciona a aparência do controle. Aqui você encontrará (veja na Fig 88) o visual de como ficará o controle e definirá certas características:

- **Auto preenchimento ao digitar**: Ativa ou desativa o auto preenchimento para os itens existentes na primeira coluna do controle. Opção útil para listas muito grandes.

- **Ordenada**: Ativa ou desativa a classificação automática para os itens existentes na primeira coluna do controle.
- **Largura da Lista**: Define a largura da lista.

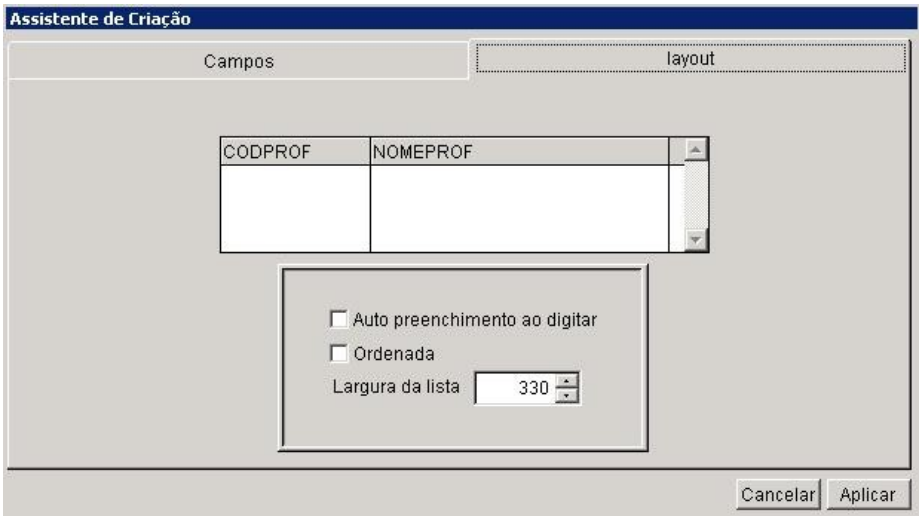


Fig 88 Layout do controle.

- Selecione **Auto preenchimento...**
- Clique no botão esquerdo do mouse em **Aplicar** para salvar as modificações.

Nota

Caso –posteriormente- você precise acessar a janela **Assistente de Criação** para fazer alguma modificação no controle já incluído, basta abrir o formulário da entidade pelo Visual Kit 5 e clicar o botão direito do mouse no controle. Aparecerá o menu de contexto como é exibido na Fig 89. Escolha a opção **Editar** e você acessará novamente o **Assistente de Criação.**

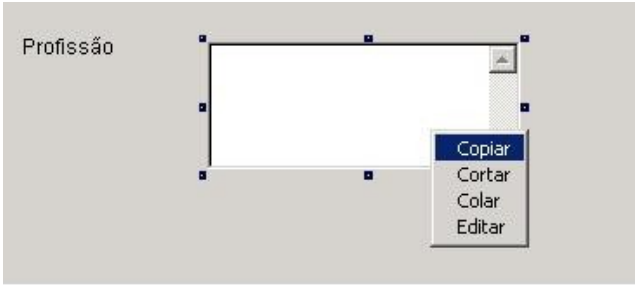


Fig 89 Menu de contexto para o ListBox

Vejamos como ficou nosso primeiro sistema com entidades relacionadas. Vamos a executá-lo da mesma maneira como foi explicado no Capítulo IV quando criamos o primeiro sistema.

Primeiramente, abra a entidade *Profissão* para cadastrar várias profissões. Cadastre várias profissões. Na Fig 90 aparece uma consulta feita após o cadastramento de algumas profissões a maneira de exemplo.

Fig 90 Profissões cadastradas.

Inclua agora diferentes clientes e veja como funciona o relacionamento entre as entidades através do controle *ListBox*. O formulário dos clientes ficou como mostra a Fig 91. Ele possui uma lista para seleção, com as profissões recém cadastradas no passo anterior...Bonito eh!...

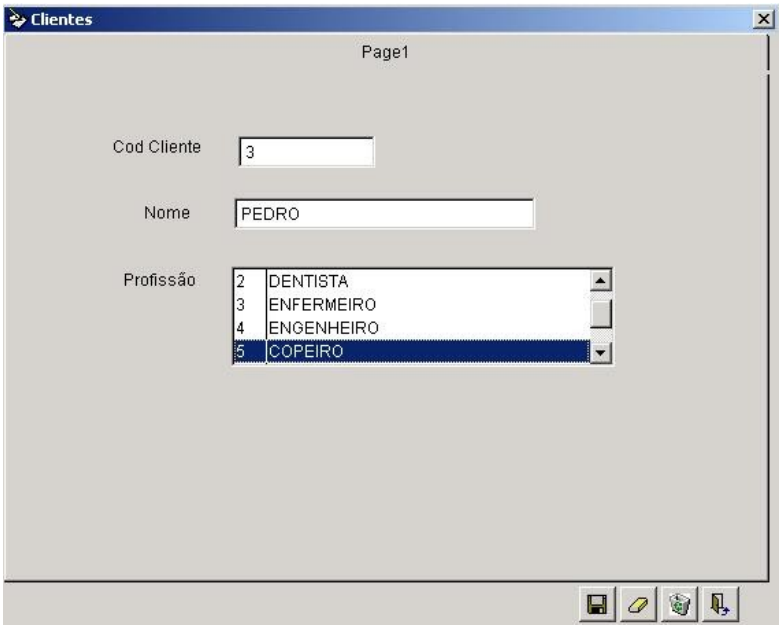


Fig 91 Formulários de Clientes com *ListBox*

Cadastre vários clientes, use nome de amigos, namoradas, familiares, etc... Vamos dar uma olhada usando a operação de *Consulta* e ver como

estão sendo gravados os dados?. A primeira e a segunda coluna gravarão o código e o nome do cliente respectivamente, já a terceira coluna – referente ao controle *ListBox*- ficou com o código da chave das profissões. Veja na Fig 92.

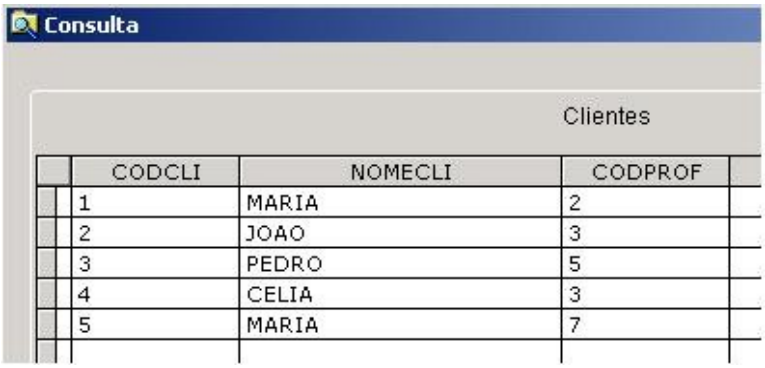


Fig 92 Consulta da Entidade *Clientes*.

Novo *ComboBox*:

Fig 93 Visual do controle *ComboBox*

O *ComboBox* combina uma caixa de texto (*TextBox*) com uma listagem de opções (*ListBox*). Como o *ListBox*, o *ComboBox* permite exibir uma lista de opções para o usuário, mas com a seguinte diferença:

- No *ListBox* o usuário deverá sempre dar um clique no item desejado e, no *ComboBox*, o usuário poderá tanto selecioná-lo quanto digitá-lo no *TextBox* por ele apresentado.
- Um *ListBox* sempre apresentará uma porção da lista, estará sempre aberto ocupando mais espaço no formulário. O *ComboBox* poderá ser apresentado fechado.

- Você deverá utilizar o *ComboBox* sempre que quiser exibir uma lista de opções ao usuário e não houver muito espaço livre em sua interface, ou caso você deseje oferecer duas formas de o usuário especificar o item desejado: selecionado-o ou digitando-o.

O *ComboBox* igual que o *ListBox* precisa de uma outra entidade que controlará e armazenará a listagem das opções. Portanto é um controle que precisa estar relacionado.

Atenção:
Tanto o controle *ComboBox* como *ListBox* gravarão na tabela da entidade atual o campo chave da tabela da entidade relacionada. Inclusive quando esta não for selecionada para ser exibida no próprio controle.

Os passos no desenho do *ComboBox* são muito similares aos descritos para o *ListBox*. Para ver como funciona, vamos criar um sistema idêntico ao anterior, substituindo o controle *ListBox* pelo *ComboBox* . Tudo é feito da mesma maneira, portanto vamos pular as partes iguais.

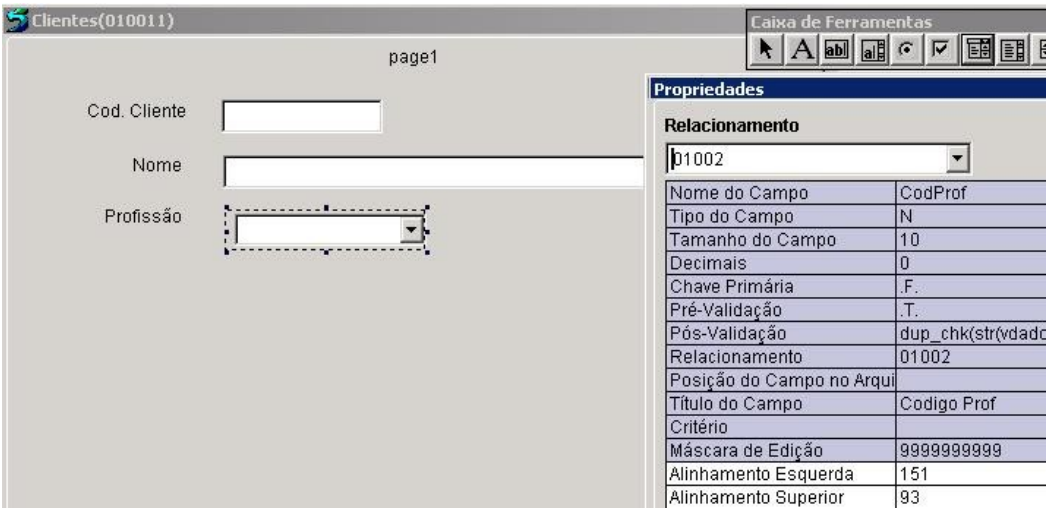


Fig 94 Colocando o controle *ComboBox*

- Selecione o controle *ComboBox* e insira-o no formulário.
- Selecione como entidade relacionada **Profissões;**
- Aparecerá o **Assistente de Criação** como no *ListBox*
- Você selecionará primeiramente os campos e depois definirá o *layout* do controle.

Já a pagina do *layout* (veja na Fig 95) se difere um pouco. Aqui você definirá as seguintes características:

- **Auto preenchimento ao digitar**: Ativa ou desativa o auto preenchimento para os itens existentes na primeira coluna do controle. Opção útil para listas muito grandes.
- **Igualar a largura com a lista**: Ativa ou desativa a opção de igualar a largura do controle à largura da lista dos itens.
- **Largura da Lista**: Define a largura da lista.
- **Campo de retorno na tela:** Define qual campo ficará exibido na tela depois que é feita a seleção pelo usuário. Note que não é exatamente o campo que ficará gravado internamente na entidade. O campo que será gravado na tabela associada à entidade é o campo chave da entidade relacionada.

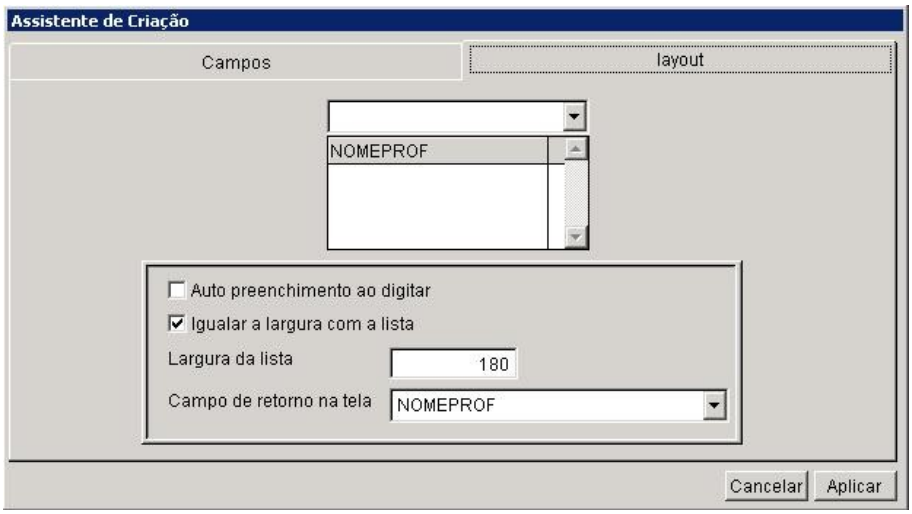


Fig 95 Layout do *ComboBox*

Vejamos como funciona este controle. Abra o sistema gerado e selecione a entidade Clientes.

- Inclua um conteúdo para os campos Cód Cliente e Nome;
- Clique no controle *ComboBox* e aparecerá a lista de profissões cadastrada na entidade Profissões (veja na Fig 96);
- Selecione uma profissão;
- O controle é fechado novamente como mostra a Fig 97.

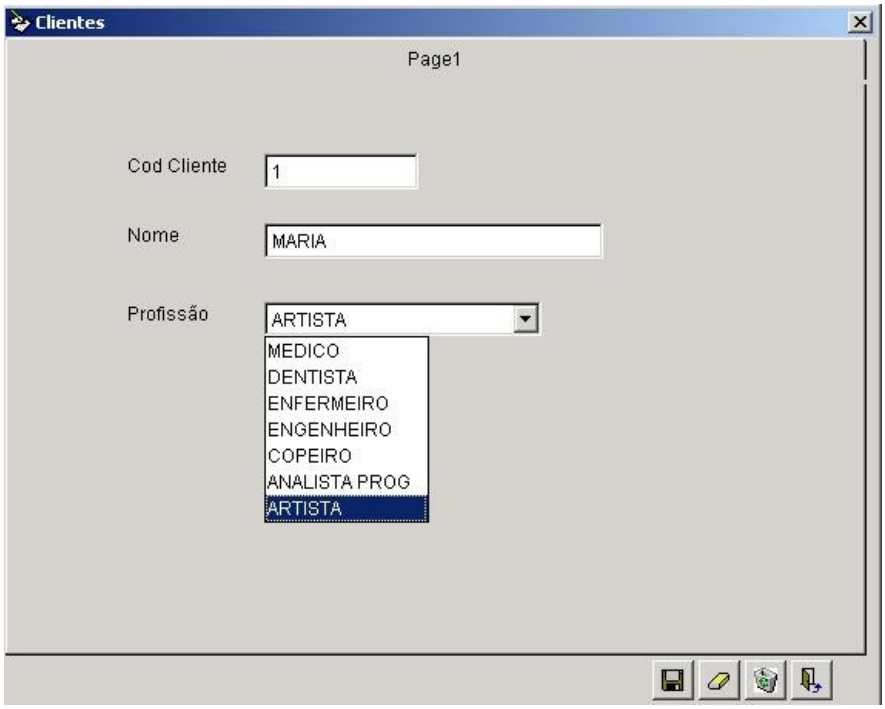


Fig 96 Cadastrando *Clientes* usando o *ComboBox*

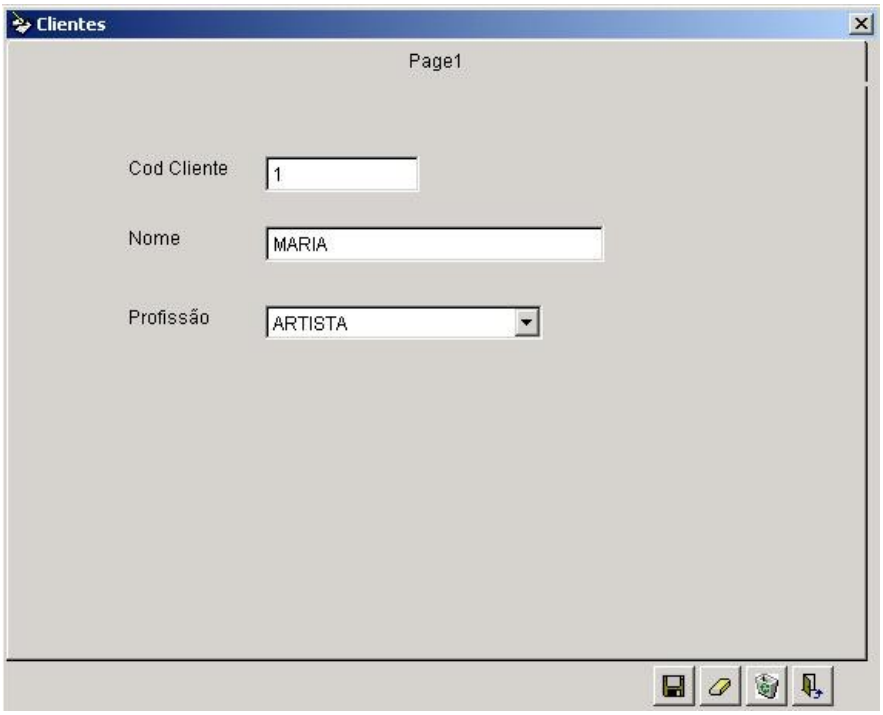


Fig 97 No final da seleção o *ComboBox* fecha novamente.

Para ver como foram gravados os dados internamente clique em **Consulta.** Os dados gravados terão o mesmo formato mostrado na Fig 92.

Grid

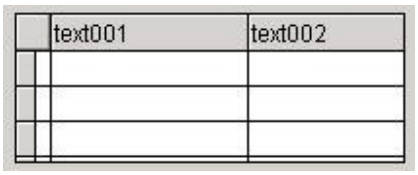


Fig 98 Controle *Grid* –grade

O controle *Grid* ou grade dará um acabamento profissional aos formulários criados por você. Ele permite a apresentação de dados num formato tabular linhas x colunas, semelhantes a uma planilha. Inserido no formulário de uma entidade, as colunas do *Grid* exibirão o conteúdo de outros controles da mesma entidade ou de entidades relacionadas.

Para definir as características do controle na etapa de desenho, você usará tanto a janela de propriedades como um assistente de criação que descreveremos seguidamente.

Internamente o objeto *Grid* não criará campos na tabela da entidade, servindo unicamente como uma consulta dentro do formulário.

Atenção:
Para inserir um controle *Grid* no formulário de uma entidade, é preciso que a chave primária da entidade seja chave composta, ou seja, esteja formada por mais de um campo.

Algumas das propriedades particulares deste objeto são:

PROPRIEDADE	DESCRIÇÃO
Barras de Rolagem	Sempre que você trabalha com múltiplas linhas e/ou colunas deve ativar a barra de rolagem, permitindo que o usuário possa percorrer todas as linhas e/ou colunas. Valores: 0- Nenhuma 1- Horizontal 2- Vertical 3- Ambos
Exibe Linhas na Grade	Exibe as linha horizontais e verticais na grade. 0- Nenhuma

	1- Horizontal 2- Vertical 3- Ambos
Largura da Linha	Se refere à largura das linhas da grade
Cor da Linha	Especifica a cor a ser utilizada para as linhas da grade
Altura da Linha	Alturas da linhas na grade

Vejamos no seguinte exemplo como funciona este controle.

Exemplo de entidade com chave composta.

A proposta: Criar um sistema simples (uma única entidade) para cadastrar os jogadores de futevôlei da Copa Mundial referenciando pelo nome do equipe e o número da camisa do jogador.

Neste caso a chave da entidade está composta pelos campos “Time” e “Camisa”. Na entidade, o nome do clube poderá aparecer várias vezes, igual ao número da camisa, mas a combinação Time-Camisa é única: não pode haver dois jogadores com a mesma camisa no mesmo time.

Fig 99 Estabelecendo a estrutura do sistema

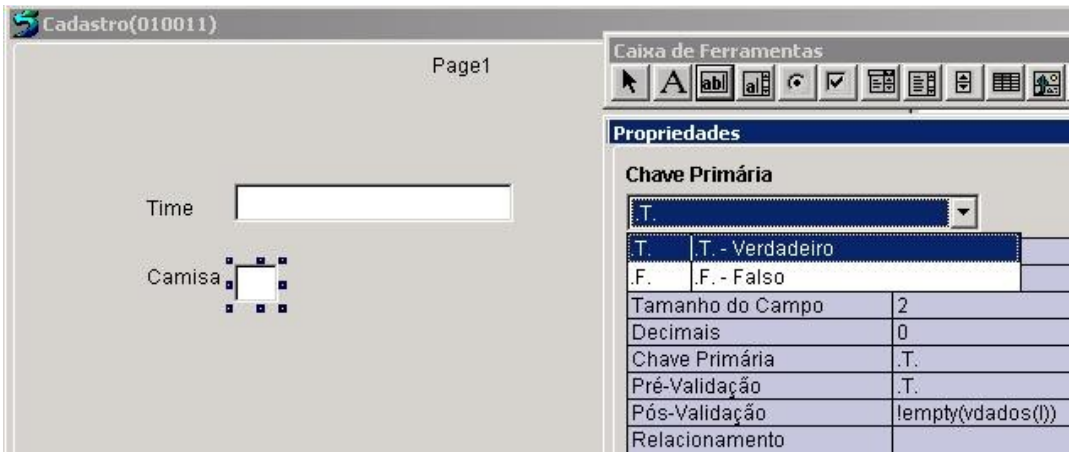


Fig 100 O segundo campo deverá ser definido como chave.

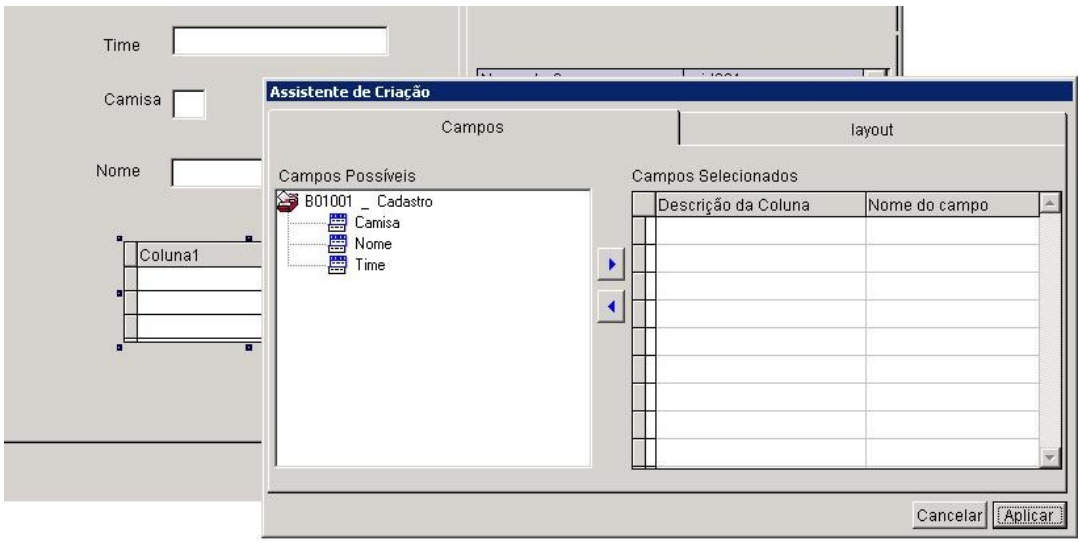


Fig 101 Assistente de Criação

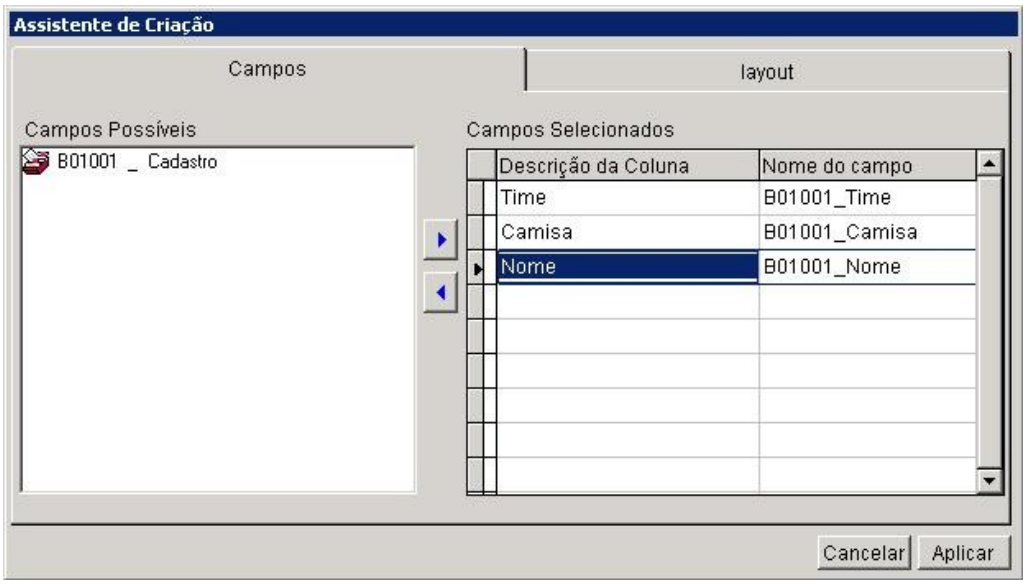


Fig 102 Escolha dos campos para exibição no objeto

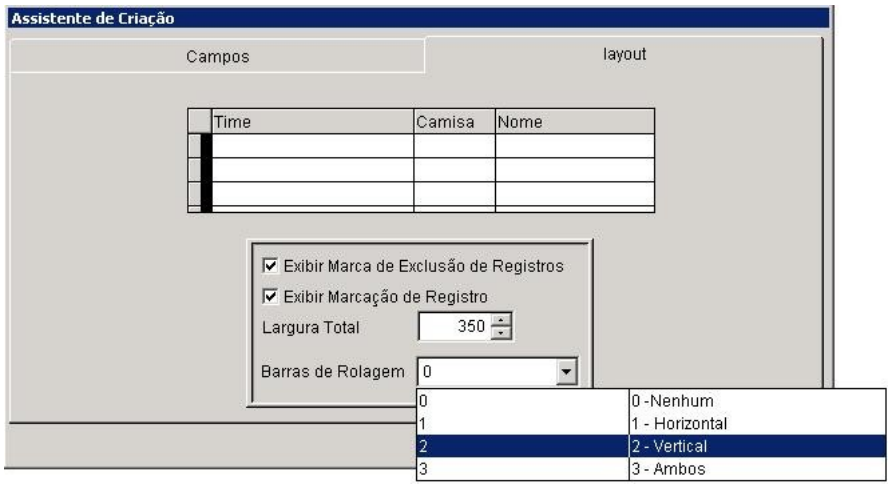


Fig 103 Página de *Layout* no Assistente de Criação

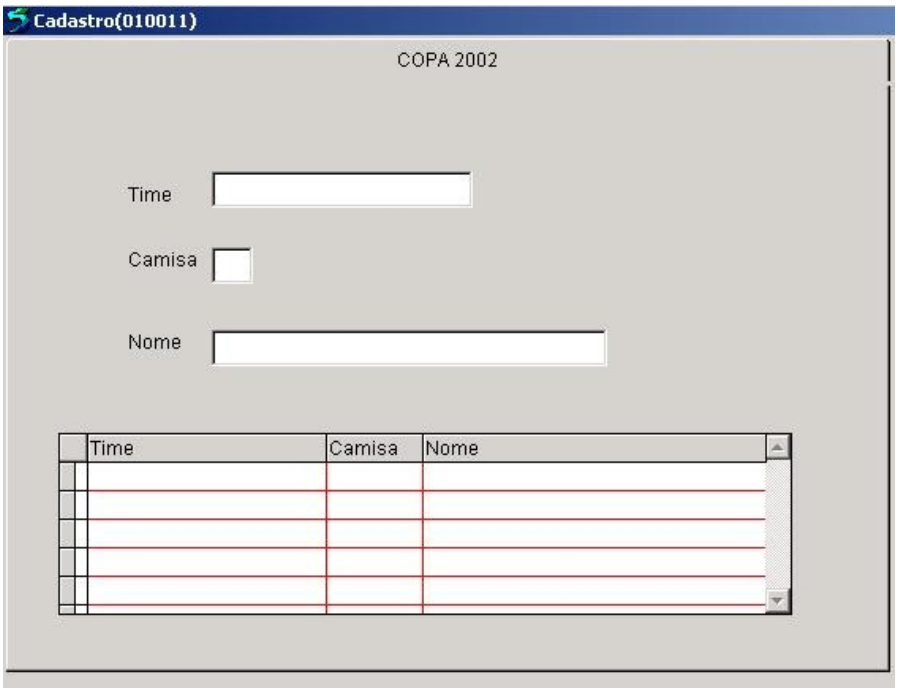


Fig 104 Visual final do formulário

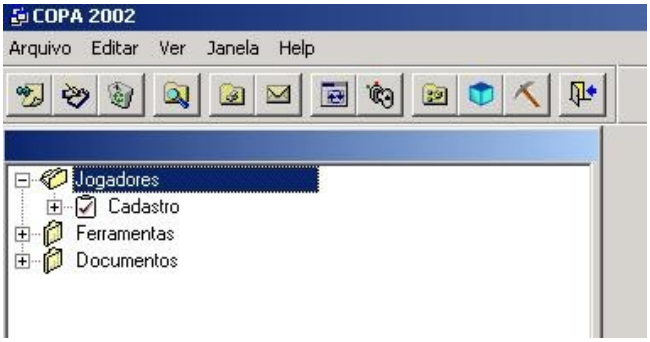


Fig 105 Sistema Gerado

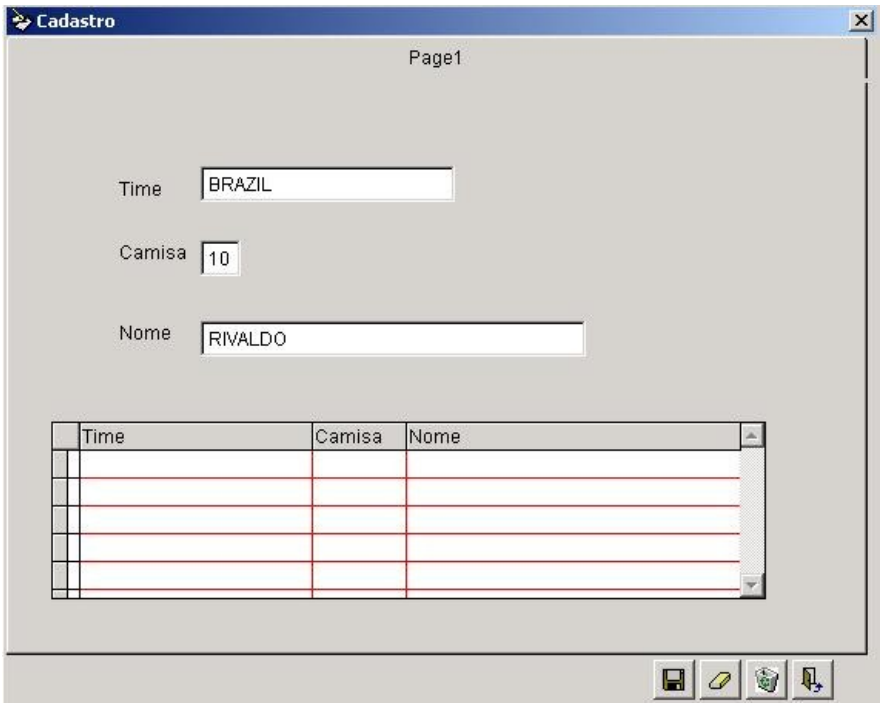


Fig 106 Acessando o formulário do sistema gerado.

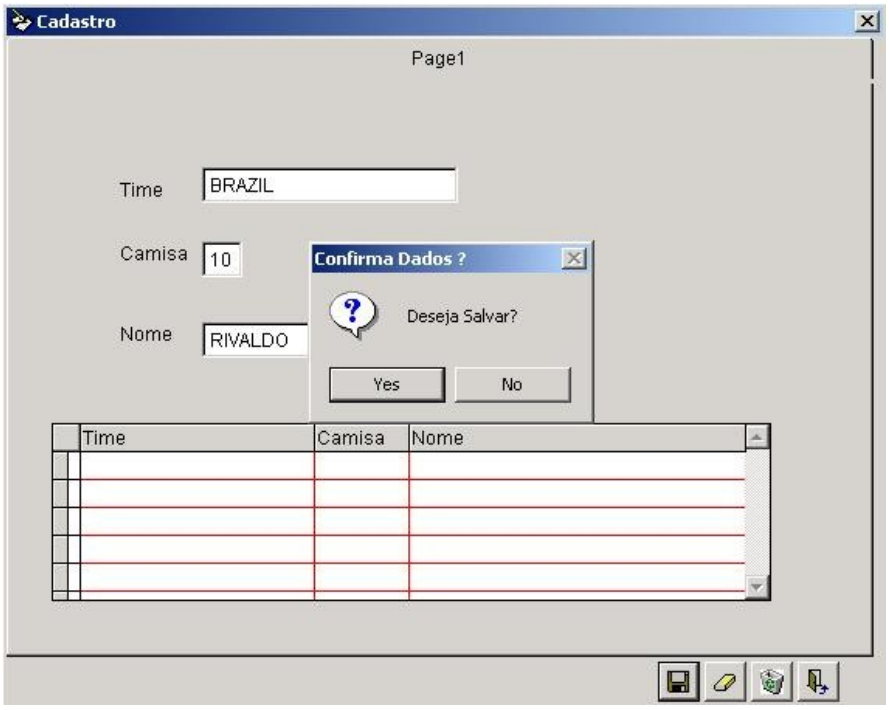


Fig 107 Inserindo o primeiro registro

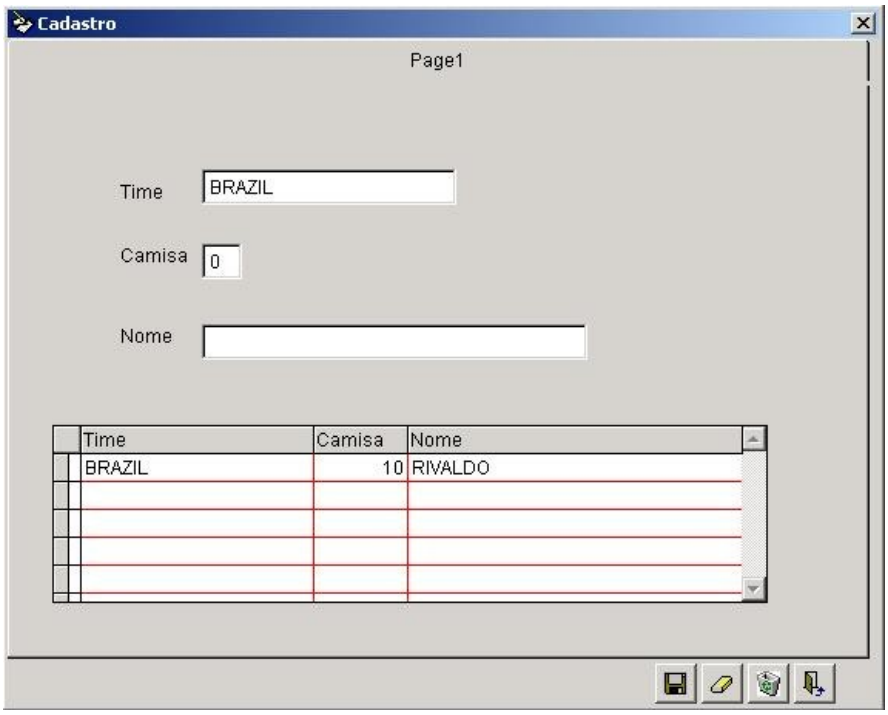


Fig 108 Inserindo o segundo registro

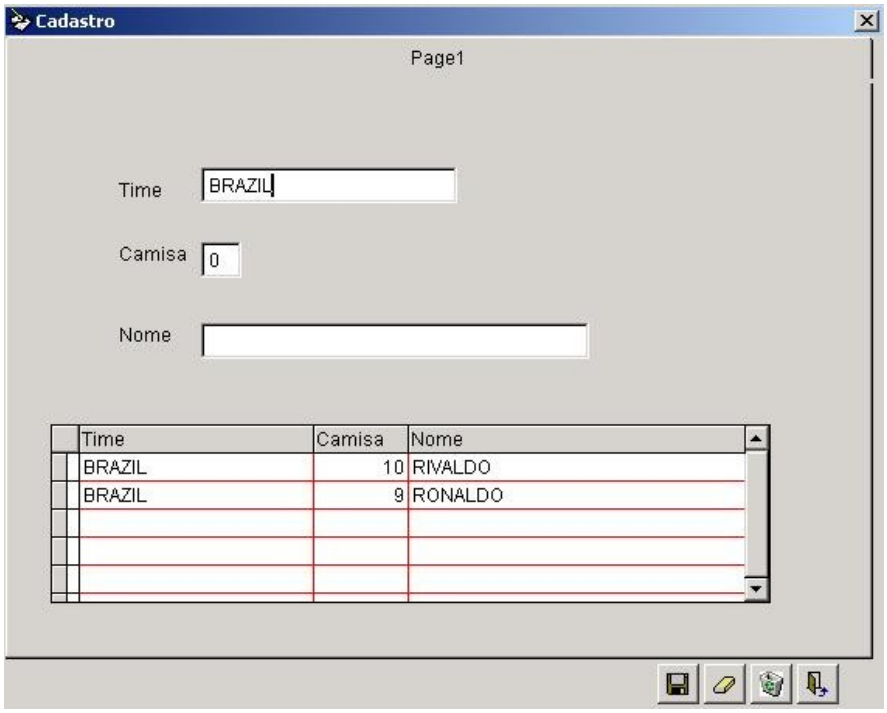


Fig 109 Segundo registro inserido

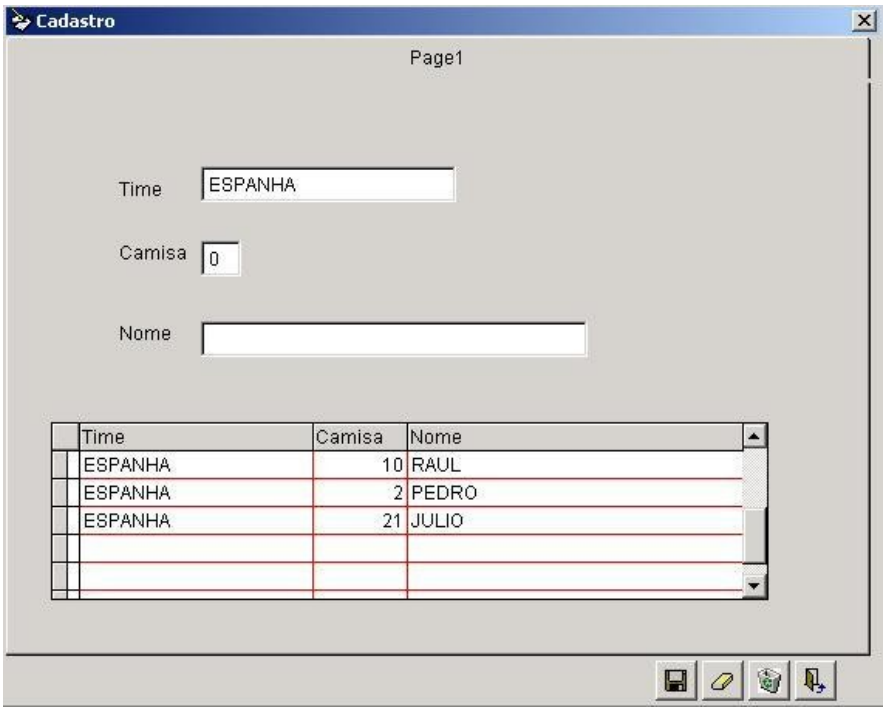


Fig 110 Inserindo registros com a primeira chave diferente

Fig 111 Resultado das inserções.

Botão

Fig 112 O controle botão.

PROPRIEDADE	DESCRIÇÃO
Ao Clicar	Comando a executar quando é realizado um clique sobre o botão
Ao Clicar Duas Vezes	Comando a executar quando é realizado um clique duplo sobre o botão
Figura	Imagem que aparecerá no botão
Descrição do Botão	Texto que aparecerá em cima do botão

Exibe

Fig 113 Controle Exibe

Picture

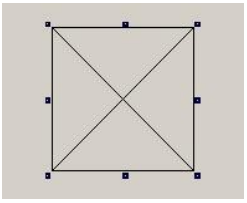


Fig 114 O controle *Picture*

Calendario

Fig 115

Link

Fig 116

Criando novas funcionalidades usando os eventos.

Neste seção abordaremos os recursos que permitem colocar novas funcionalidades aos controles descritos anteriormente. Estas funcionalidades estão associadas às ações que o usuário do sistema gerado realizará sobre determinados controles.

Atenção!

A *Designer-Pro* não utiliza as opções de Pré e Pós Validação descrita no capítulo anterior. Estas funcionalidades são realizadas através das ações dos eventos que descreveremos nesta seção.

Entendendo os eventos

No desenvolvimento de sistemas, podemos definir que os “eventos” são ações reconhecidas pelos controles. As ações podem ser: clicar o botão direito do mouse em cima de um controle *Botão* ou usar a tecla TAB fazendo que um determinado *TextBox* perca ou ganhe o foco, etc.

Cada controle disponível no Visual Kit 5 possui um conjunto de eventos a ele relacionado, por exemplo, “Ao Clicar”, “Ao Clicar Duas Vezes”, “Ao receber o foco”, etc. É comum encontrar eventos que estão disponíveis em vários objetos diferentes. Veja na janela de propriedades da Fig 117 onde é que estão localizados tais eventos.

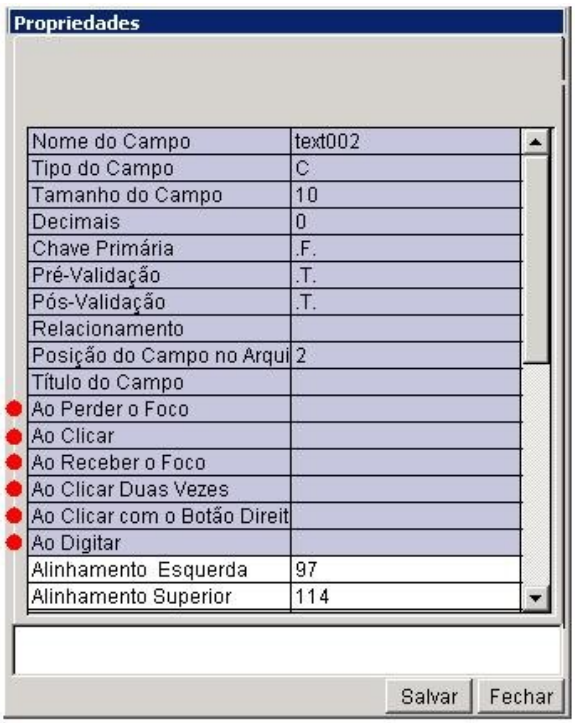


Fig 117 Localização dos eventos na janela de propriedades.

Bom, onde é que você se encaixa nesta história? Pois você poderá definir a “ação” do seu sistema a tais eventos. O Visual Kit 5 possui dois úteis recursos – o Editor de Macros e Editor de Código - que permitirão definir as funções que você quer que o sistema execute quando acontecer um evento registrado. Por exemplo: exibir uma mensagem, cálculos diversos, colocar todo conteúdo em maiúsculas, aceitar um valor numérico num determinado intervalo, etc.

O que torna a ferramenta de desenvolvimento Visual Kit 5 tão poderosa é a profusão de controles possíveis, suas propriedades, os eventos que cada um deles pode responder e os procedimentos que podem ser definidos. Mas, você deve ir aprendendo aos poucos, à medida que você vai precisando usar esse ou aquele controle, propriedade ou evento.

Consideremos algumas definições importantes.

Evento: É um acontecimento no ambiente de execução, reconhecido por um controle para o qual você pode definir uma ação. O exemplo clássico de um evento é o clique do mouse. Você controla o comportamento do controle no momento em que ele detecta o evento –o clique do mouse- , escrevendo uma ação ou procedimento.

Ações: São procedimentos que um controle é capaz de executar. Os controles já incluem procedimentos característicos de seu funcionamento, mas você pode criar e acrescentar outros. Os procedimentos são funções que ficam ligadas aos controles. Por exemplo, podemos criar uma função associada ao evento **Ao Clicar** de um controle *TextBox*, que exibirá uma janela com uma determinada mensagem. No Visual Kit 5 estas funções são definidas usando o editor de macros ou o editor de códigos, que veremos a seguir.

Macros: São procedimentos criados a partir de funções preestabelecidas pelo Visual Kit 5 e disponibilizadas para que o usuário possa usá-las de maneira simples.

Códigos: O Visual Kit 5 disponibiliza um editor de código na linguagem FoxPro para a criação dos procedimentos dos eventos. Este código será compilado pelo próprio Visual Kit 5, e possui todas as funcionalidades desta linguagem.

Foco: Diz-se que um controle em um formulário ganha ou perde o foco, quando ele se torna ou deixa de ser o objeto atualmente selecionado, seja por ação do usuário ou programaticamente. Esses eventos são temporariamente vizinhos. A maioria dos controles possuem o evento Receber e Perder foco.

Os principais eventos

Ao desenvolver sistemas com o Visual Kit 5, você trabalha com os eventos disponíveis na janela de propriedades. Você não pode criar, modificar ou eliminar eventos, mas se pode escrever as funções do evento que é executado quando eles ocorrem.

A lista de eventos mais comuns no Visual Kit 5, e que são reconhecidos pela maioria dos controles está descrita na seguinte tabela:

Evento	Quando acontece
Ao Clicar	Usuário clica o botão primário do mouse.

Ao Clicar Duas Vezes	Usuário dá uma dupla-clicada sobre o controle.
Ao Clicar com o botão direito	Usuário clica com o botão secundário.
Ao Perder o Foco	O controle “perde o foco”, seja por ação do usuário usando o mouse ou teclado (TAB).
Ao Receber o Foco	O controle “recebe o foco”, seja por ação do usuário usando o mouse ou teclado (TAB).
Ao Digitar	Usuário pressiona tecla para digitar.

Para definir os procedimentos para uma determinada ação você deve clicar em cima do evento específico e escolher qual dos recursos usar para a edição ou criação da funcionalidade: *Expression*, *Macro* ou *Code*?. Veja na Fig 118.

Fig 118 Selecionando o modo de edição do procedimento dos eventos

O Editor de Expressões

Usando o Editor de Expressões você poderá criar todo tipo de expressões, tanto numéricas como de caractere. Você poderá manipular não somente o conteúdo ou valor do campo, mas também todas as restantes características visuais como posição dentro do formulário, cor e formato

de fonte, visibilidade, etc. Poderá criar também expressões lógicas de tipo *SE – ENTÃO*.

Expressões matemáticas

Para mostrar que é muito fácil implementar expressões em geral usando este editor, vejamos o seguinte exemplo. Vamos criar um formulário composto por três campos mais o campo chave, e onde o conteúdo do terceiro campo seja a soma dos dois restantes. Usaremos o evento **Ao Receber o Foco** do último campo para executar automaticamente a soma.

Depois de desenhar o formulário como mostra a Fig 119 siga os seguintes procedimentos para implementar a soma.

- Clique sobre o campo que receberá a soma e selecione o evento **Ao Receber o Foco.**
- Será aberta a janela da Fig 120 .

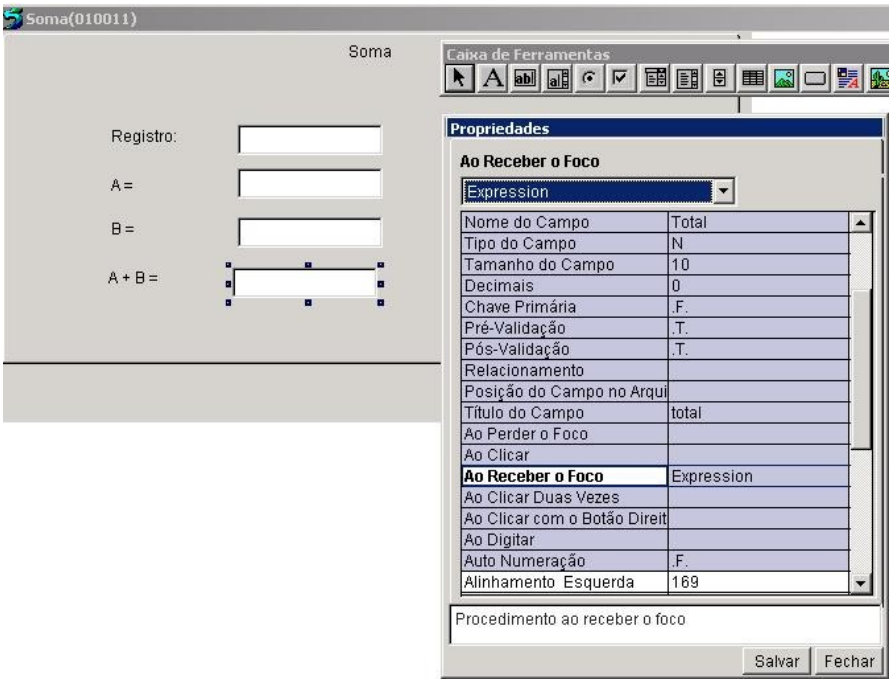


Fig 119 Chamando o Editor de Expressões para o evento Ao Receber o Foco

Esta janela é composta por duas paginas: a pagina de criação de expressões simples e a pagina de expressões com condições ou lógicas. Como nosso exemplo é uma simples soma sem avaliação de condições usaremos a primeira pagina.

A pagina de expressões é composta por quatro quadros: um superior onde ficara gravada a expressão propriamente dite e três quadros inferiores que ajudarão selecionar os valores que queremos manipular. No meio da janela aparece uma serie de botões com símbolos matemáticos que facilitam a elaboração da expressão.

No primeiro quadro inferior são colocados para seleção todos os controles do formulários da entidade atual e das entidades relacionadas. No segundo quadro serão exibidas as propriedades do campo selecionado e no terceiro quadro são colocados os valores predefinidos para determinadas propriedades.

Vejamos a continuação os procedimentos para conformar a expressão:

- Clique sobre o objeto relativo ao terceiro campo (Total) que receberá o valor da soma;
- Imediatamente o segundo quadro inferior mostrará todas as propriedades deste controle;
- Faça um duplo-clique sobre a propriedade **Valor do Campo;**
- Automaticamente será inserida uma expressão no quadro superior que indicará em códigos esta propriedade associada ao dado objeto do formulário;
- Clique no botão igual a “**=**” para formar a expressão tipo “C=A+B”
- Clique agora no controle Valor_1. As propriedades do quadro serão as relativas a este controle.
- Faça novamente um duplo-clique sobre a propriedade **Valor do Campo** deste controle**.**
- Clique no botão “**+**” para conformar a soma.
- Clique no segundo somando - o controle Valor_2 - e faça um duplo clique sobre **Valor do Campo**
- A expressão está conformada!!! Salve a expressão clicando sobre o botão com o ícone de disquetes.

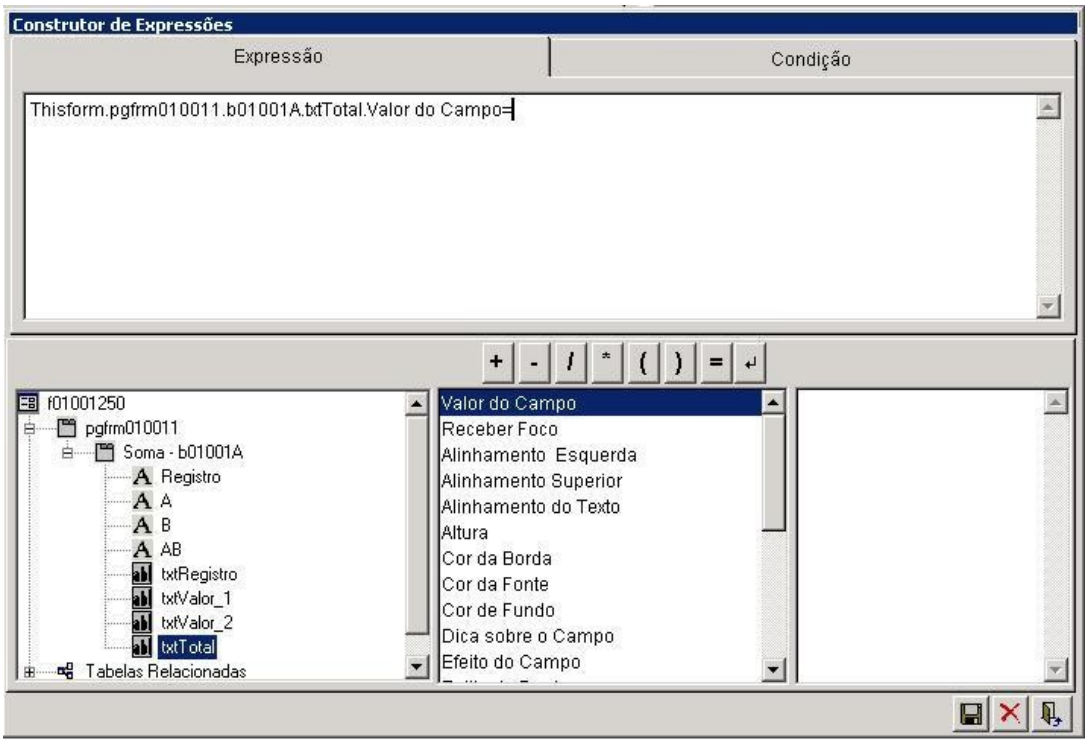


Fig 120 Editor de Expressões

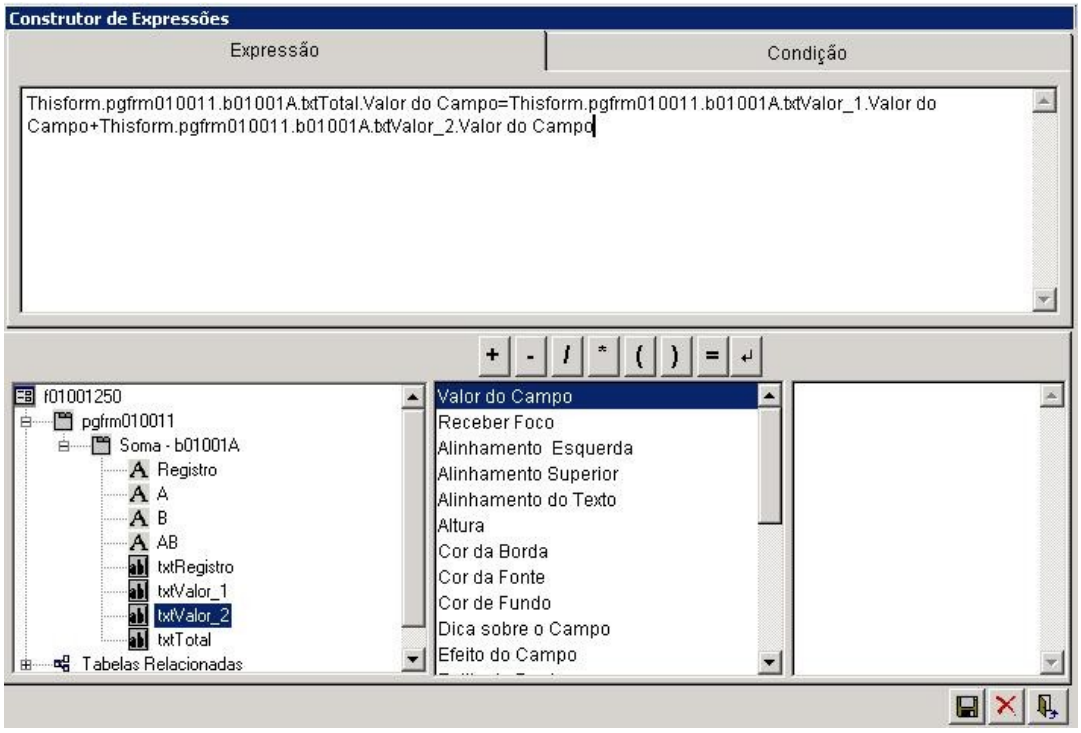


Fig 121 Expressão Conde soma conformada

Salve o sistema e abra para testá-lo. Digite o campo chave e seguidamente os valores dos elementos nos respectivos campos. Quando o ultimo campo

receba o foco (que foi o evento escolhido) deverá aparecer automaticamente o valor da soma. Veja na figura Fig 122

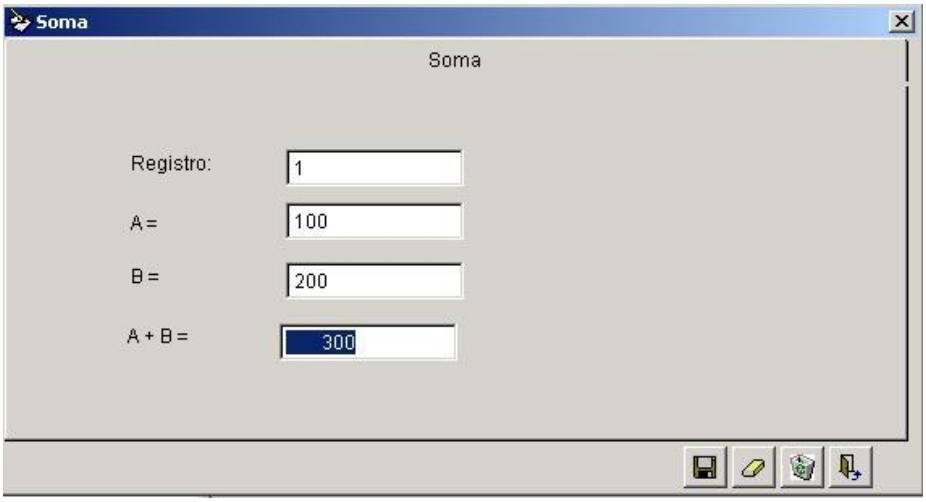


Fig 122 Testando a expressão.

O Editor de Macros.

Uma maneira simples de desenvolver as funcionalidades dos eventos dos controles é através das Macros. As Macros são procedimentos que você poderá conformar através de várias funções bases preestabelecidas pelo Visual Kit 5.

A conformação das Macros é feita por uma interface, onde você colocará – simplesmente- os dados de entrada sem necessidade de programar.

Nota:
Você poderá inclusive desenvolver macros associadas a um controle que utilize dados do conteúdo de um outro campo, faça alguma ação com eles, e retorne o resultado em terceiro outro campo.

Para que você possa entender melhor, realizaremos um pequeno teste, o qual o ajudará compreender definitivamente o que são eventos e procedimentos de eventos. Execute os passos a seguir:

- Abra um novo sistema composto por um módulo e uma entidade.
- Selecione a entidade como tipo **Formulário Externo** na operação inclusão veja na Fig 42.
- Insira dois controles *TextBox* no formulário apresentado.

Muito bem, agora imagine uma ação que o usuário executaria sobre o segundo controle, por exemplo, um clique do mouse, e quando este ocorrer, uma janela de mensagem será aberta.

- Selecione o evento **Ao clicar** do segundo controle. Veja na próxima figura.
- Selecione Macro, como forma de definir as ações do evento. Será aberta então uma janela de Seleção de Macros como mostra a Fig 124.

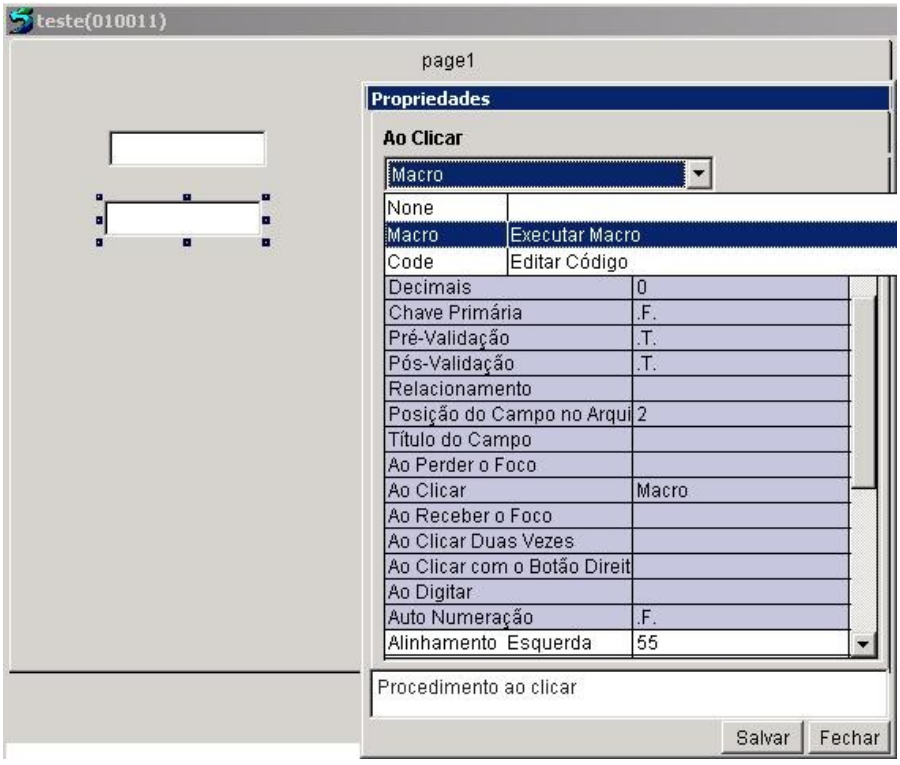


Fig 123 Inserindo dois controles e selecionando o evento Ao Clicar.

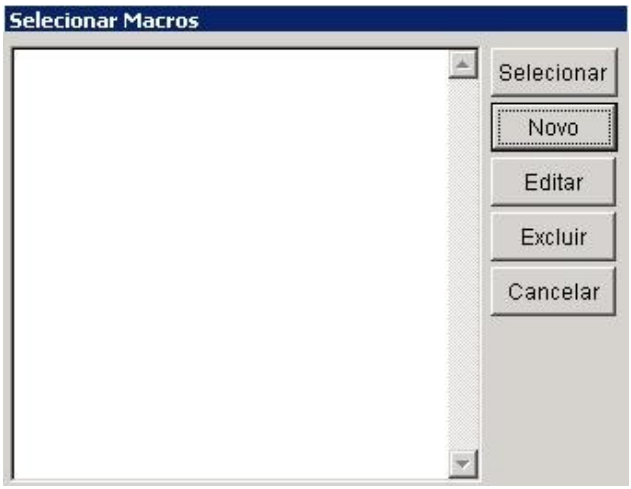


Fig 124 Janela de seleção das macros.

- Clique no botão **Novo** para cadastrar uma nova macro. Posteriormente todas as macros cadastradas estarão disponíveis numa lista para seleção nesta janela, podendo também ser reeditadas ou excluídas.
- Aparecerá neste momento, o Editor de Macros como mostra a Fig 125. Clicando na primeira linha da lista das ações, você definirá a primeira ação desta macro, selecionando uma das funções básicas (veja Fig 126). Neste exemplo criaremos uma única ação com a função **MESSAGEBOX – Caixa de Mensagem**.

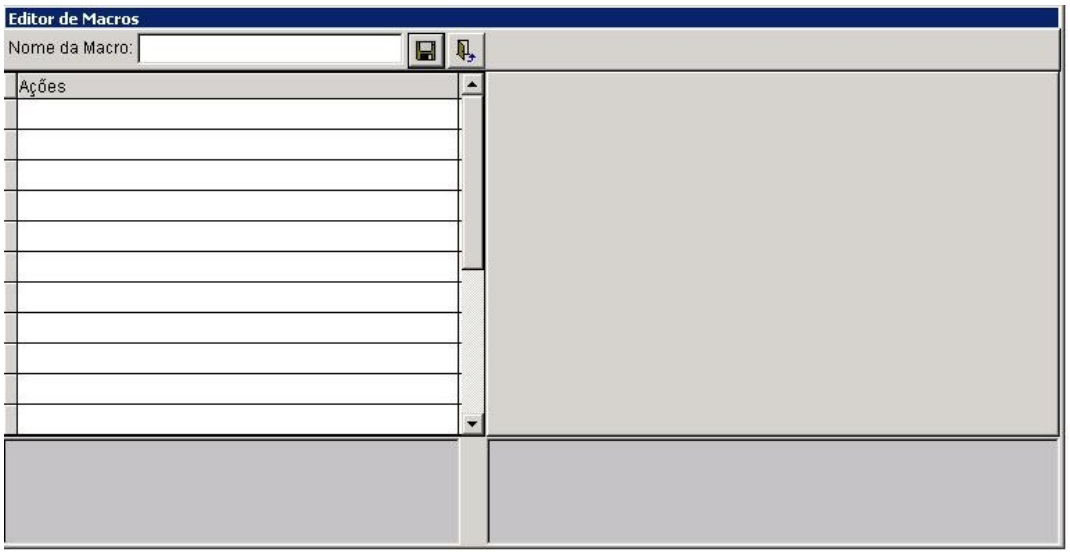


Fig 125 Janela de edição das Macros.

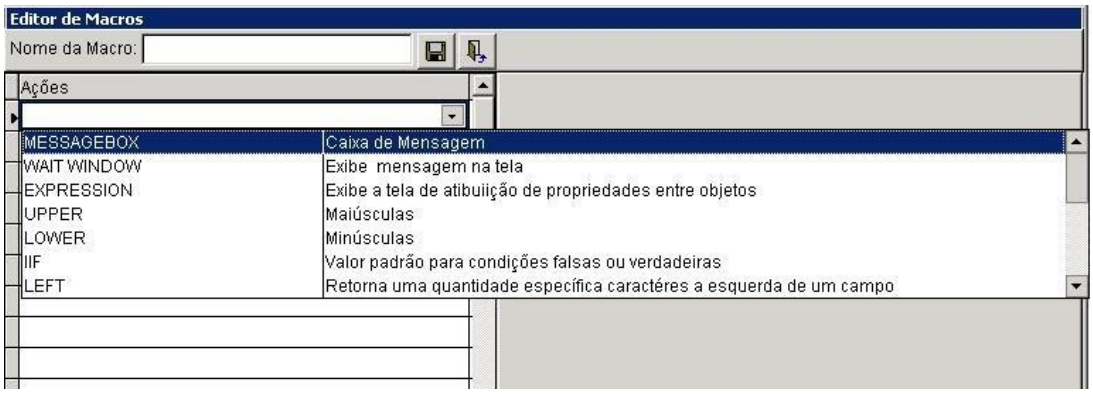


Fig 126 Selecionando as funções bases.

Nota:

Você poderá definir várias funções numa única macro, basta cadastrá-la nas outras linhas da lista de ações. A ordem de execução é definida pela ordem na lista

---

- Depois de selecionada a função **MESSAGEBOX** aparecerá no lado direito da janela uma interfase com as entradas de dados relativa a esta função. E na parte inferior um explicativo para a dada função. (Veja na Fig 127).
- Digite na primeira entrada a mensagem a ser exibida, por exemplo *Meu primeiro evento*. Na segunda entrada de dados coloque o nome do título que aparecerá na parte superior da janela que será aberta, exemplo *Testando o evento clique*.

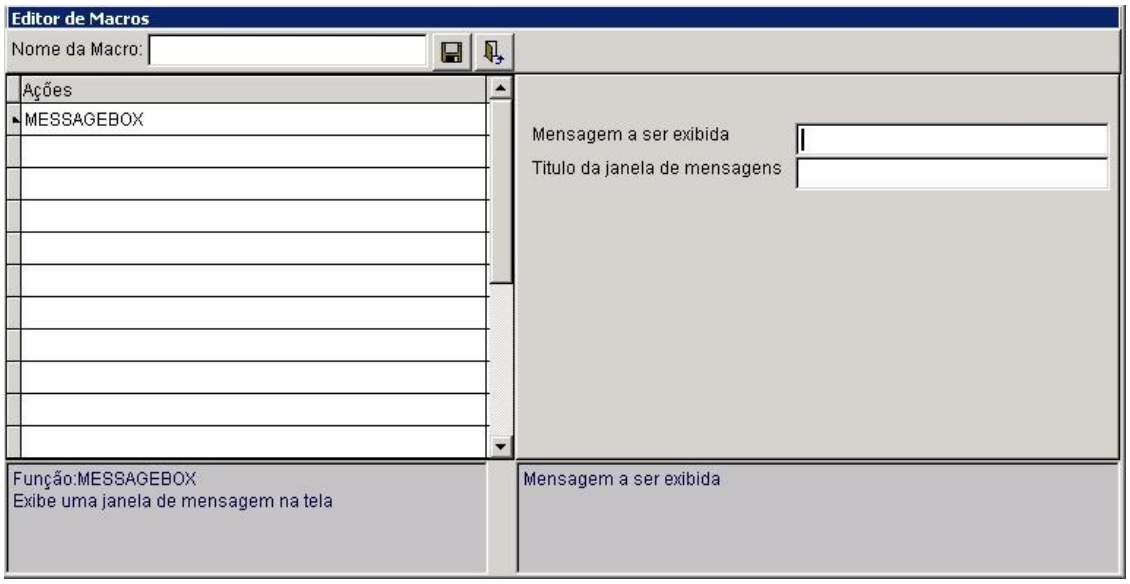


Fig 127 Função MESSAGEBOX

- Digite, por último, um nome para a macro que você acabou de definir, por exemplo **teste_clique**. Veja como ficou na seguinte figura.

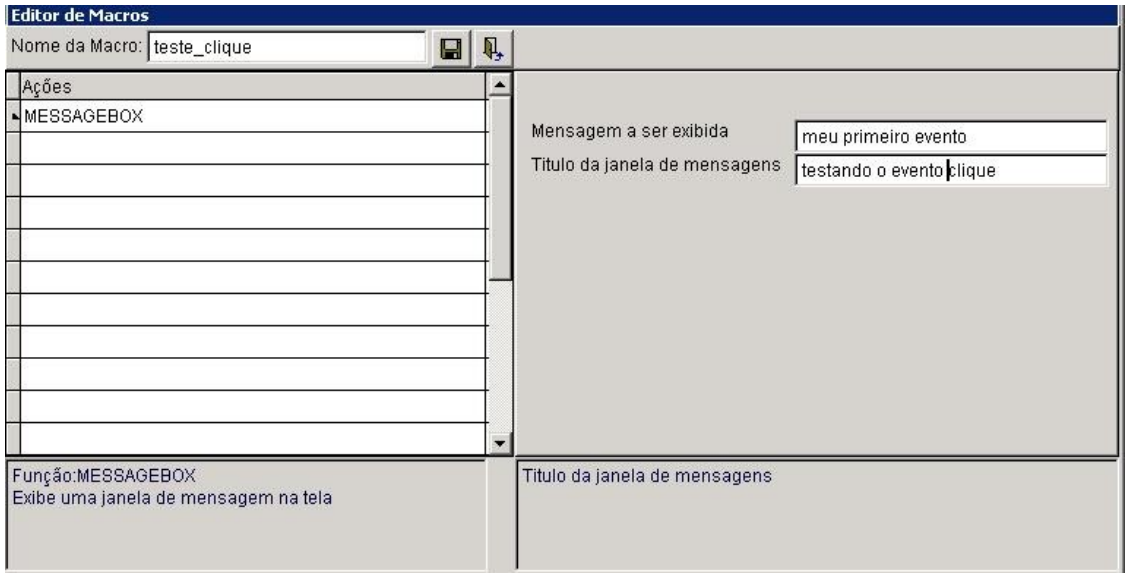


Fig 128 Definindo macro *teste_clique*

- Clique agora no botão **Salvar** (ícone disquete) e posteriormente no botão **Sair** (ícone porta). Você criou a primeira macro.
- Aparecerá – novamente- a janela de seleção de Macros, mas com a nova macro recém cadastrada por você. (Veja na próxima figura)

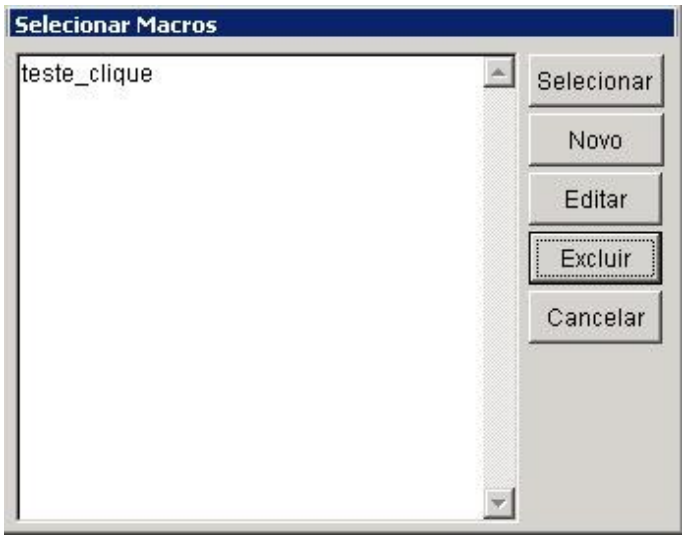


Fig 129 Macros disponíveis para seleção.

- Clique em cima da macro e clique no botão Selecionar, desta forma estamos atribuindo ao evento esta macro.
- Salve o formulário e feche o desenho do sistema. Saia do Visual Kit 5 e abra o sistema gerado neste exemplo.
- Clique sobre o segundo controle... deve aparecer uma caixa de mensagem da seguinte forma:

Fig 130 Testando o primeiro evento.

Funções bases cadastradas.

O Visual Kit 5 contém várias funções cadastradas para a elaboração das Macros. O número de funções irá aumentando em novas atualizações da ferramenta.

Exibe janela com mensagem

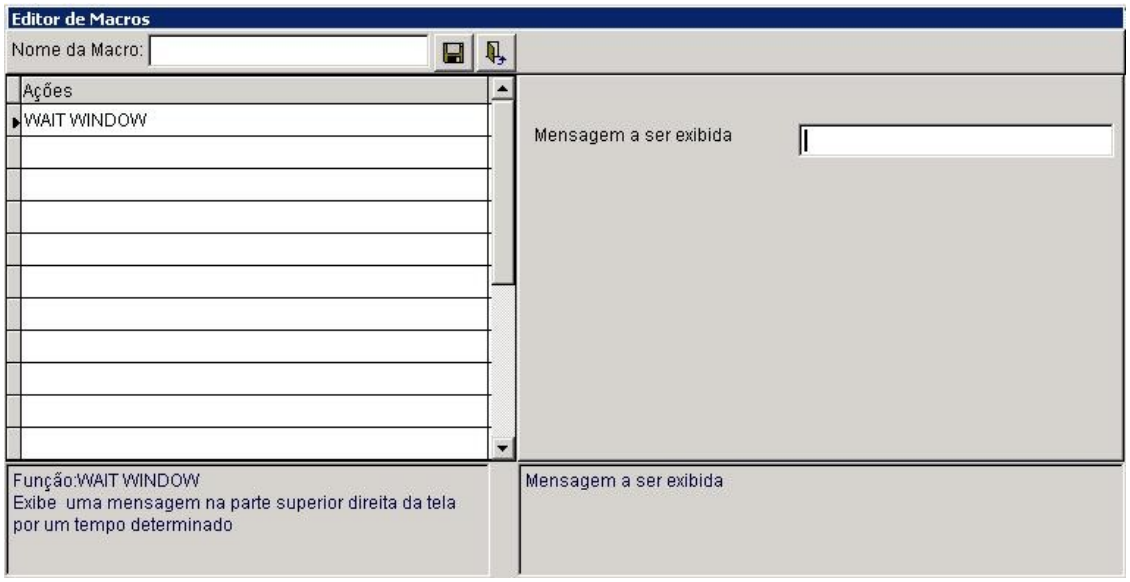


Fig 131 Função WAIT WINDOW

Construir uma expressão como macro

Usando esta funcionalidade você poderá para construir diversas expressões usando operadores descritos no Apêndice E.

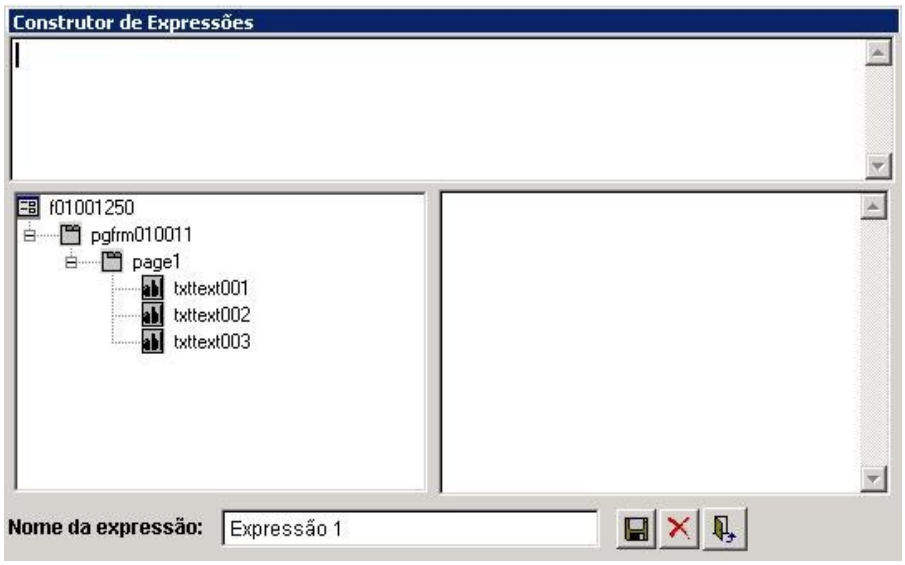


Fig 132 Criando expressões

Transforma em maiúsculas

UPPER - Retorna uma expressão específica de caracter em maiúscula. Cada letra minúscula (a – z) na expressão de caracter convertida -no retorno- em maiúscula (A – Z). Valor de retorno: Caracter

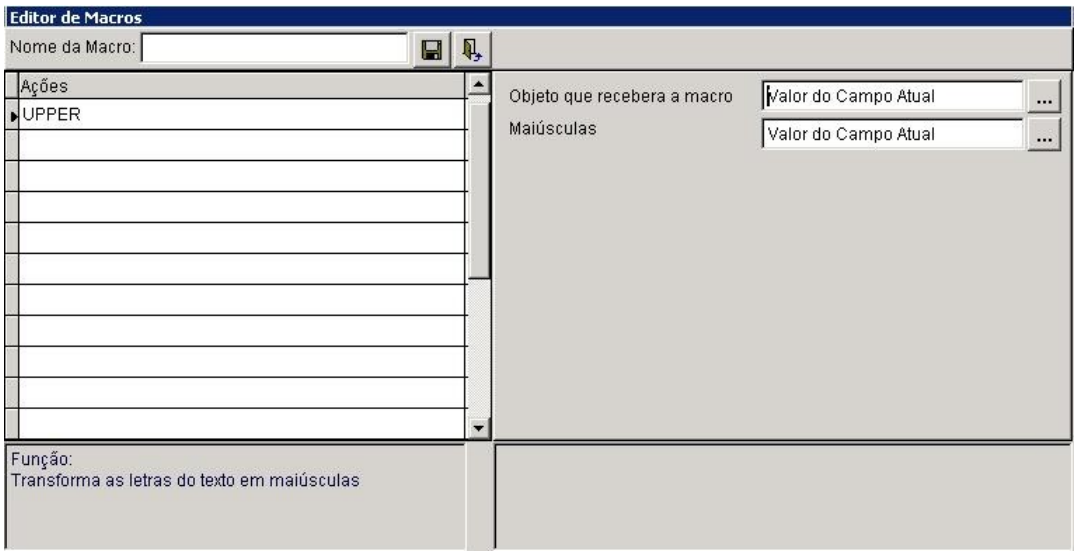


Fig 133 UPPER

A primeira entrada de dados indicada pelo rótulo “**Objeto que receberá a macro”** define o campo que receberá o resultado da macro. A segunda entrada “**Maiúsculas”** indica o campo que servirá como entrada.

Transforma em minúsculas

LOWER - Retorna uma expressão específica em caracteres minúsculos. Cada letra maiúscula (A – Z) numa expressão de caracter convertida no -retorno- em minúscula (a – z). Valor de retorno: Caracter

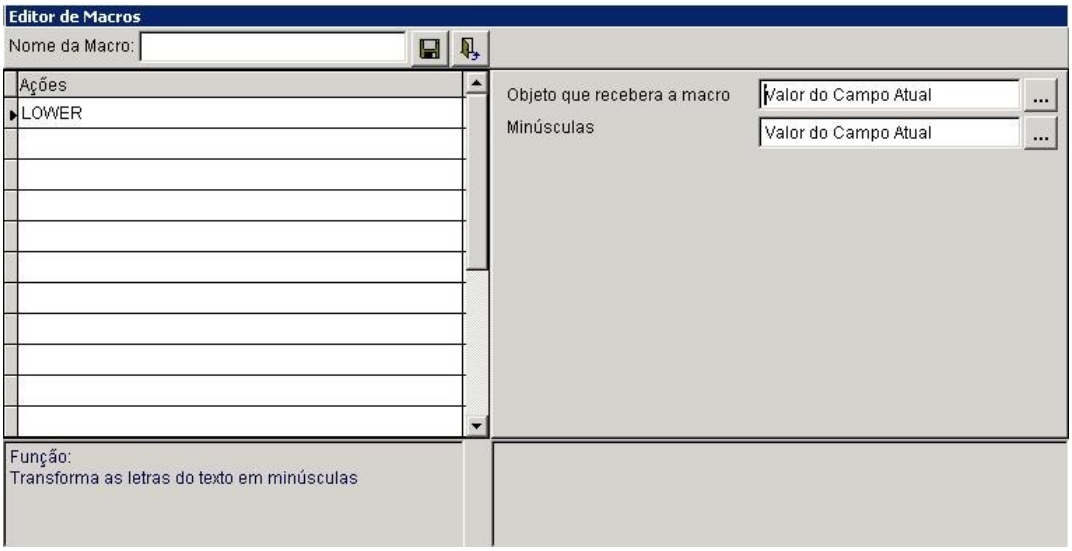


Fig 134 Função LOWER

Operador condicional

O operador *iif()*, conhecido como operador condicional poderá determinar qual função executar avaliando primeiramente uma expressão lógica. Ele retornará dois valores dependendo da expressão lógica. Valores de retorno: Caracter, Numérico e Data

Condição: Especifica a expressão lógica, ou seja, a expressão que será avaliada. Se a condição for avaliada de verdadeira (.T.), é retornado o valor da entrada **Valor para condição verdadeira**. Se a condição é avaliada de falsa (.F.), o **Valor para a condição falsa** é retornado.

O lugar de retorno é definido no primeira entrada: **Objeto que receberá a macro**.

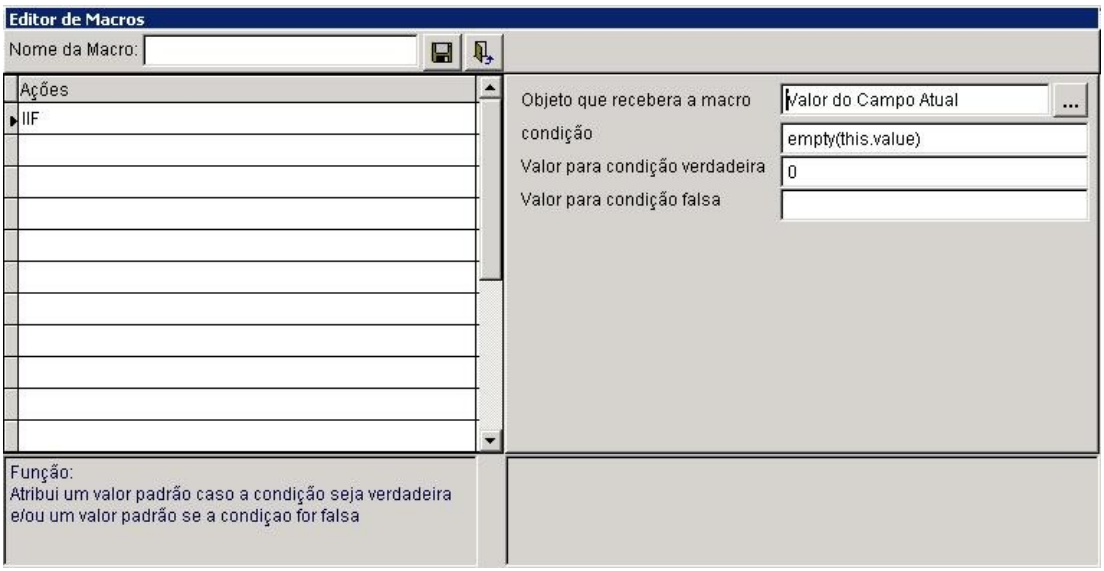


Fig 135 Função *iif*

Retorna uma quantidade específica de caracteres

LEFT: Retorna uma quantidade específica de caracteres à esquerda de uma seqüência de caracteres ou expressão *string*

RIGHT: Retorna uma quantidade específica de caracteres à direita de uma seqüência de caracteres ou expressão *string*.

Valores de retorno : Caracter

Parâmetros:
Primeira entrada de dados: Especifica o campo que receberá o retorno da função.
Segunda entrada: Especifica o campo que contém expressões em caracteres o qual os caracteres da esquerda/direita serão retornados.
Terceira entrada: Especifica o número de caracteres retornados da expressão. Esta função retornará a expressão inteira se este número é maior que a largura da expressão. Retornará vazio se este número é negativo ou 0.

Obsev.
LEFT: Os caracteres são retornados começando pelo primeiro caracter à esquerda e continuam até o número especificado.
RIGHT: Os caracteres são retornados começando pelo último caracter à direita e continuam até o número especificado.

Exemplo:

LEFT('Silva, PE', 2**)** retorna ‘Si’
RIGHT('Silva, PE', 2**)** retorna PE

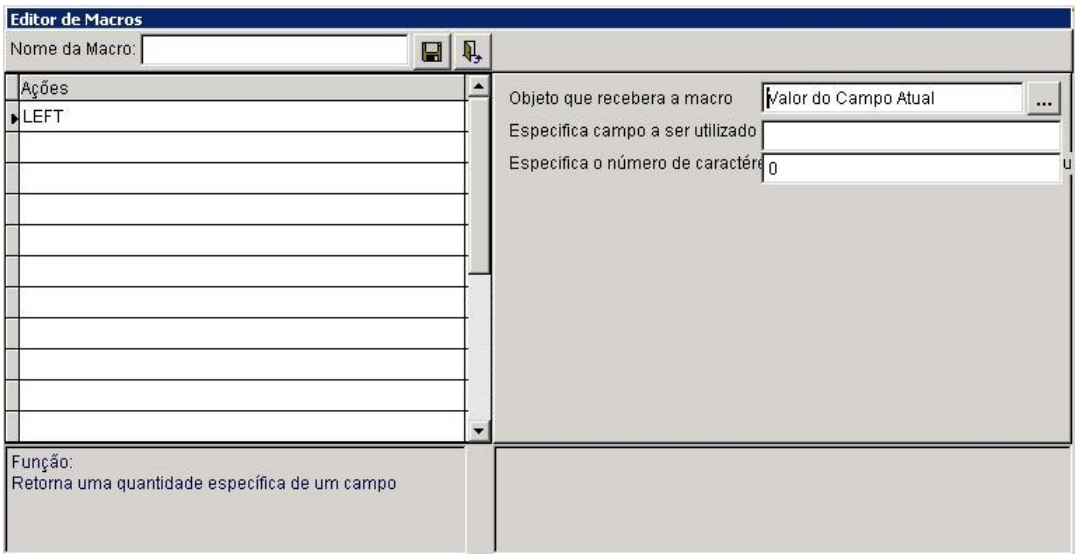


Fig 136 Função LEFT

Retira espaços em branco

Este grupo de funções retorna a expressão especificada com os espaços em branco removidos.

- LTRIM : Retira espaços à esquerda.
- RTRIM : Retira espaços à direita.
- ALLTRIM : Retira espaços à direita e esquerda.

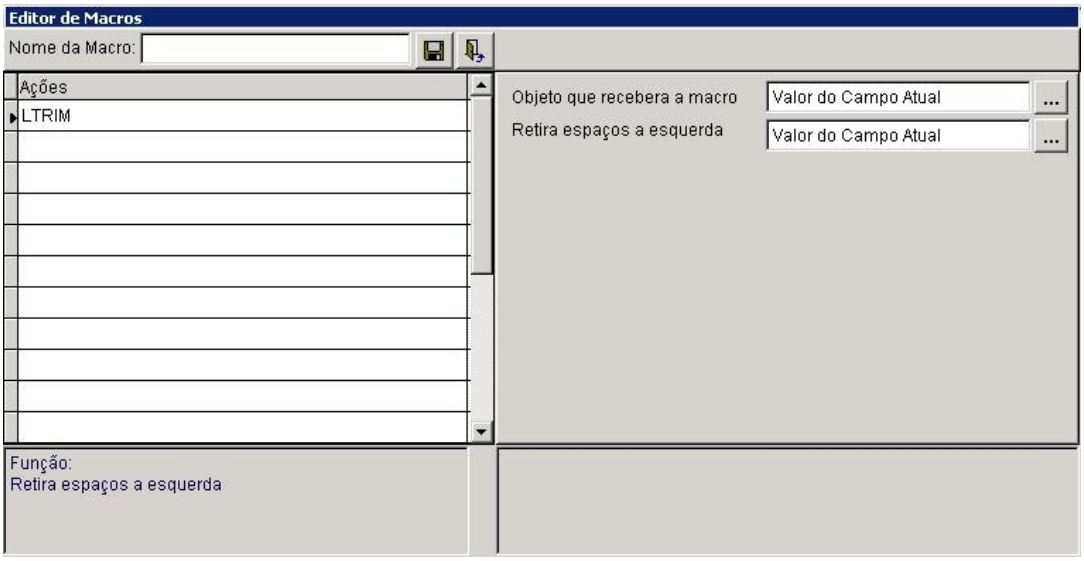


Fig 137 Função *LTRIM*

Exemplo: ‘ Rio de Janeiro ’
LTRIM: ‘Rio de Janeiro ’
RTRIM: ‘ Rio de Janeiro’
ALLTRIM: ‘Rio de Janeiro’

Retorna dia da semana

Retorna o nome do dia da semana de um campo tipo data.

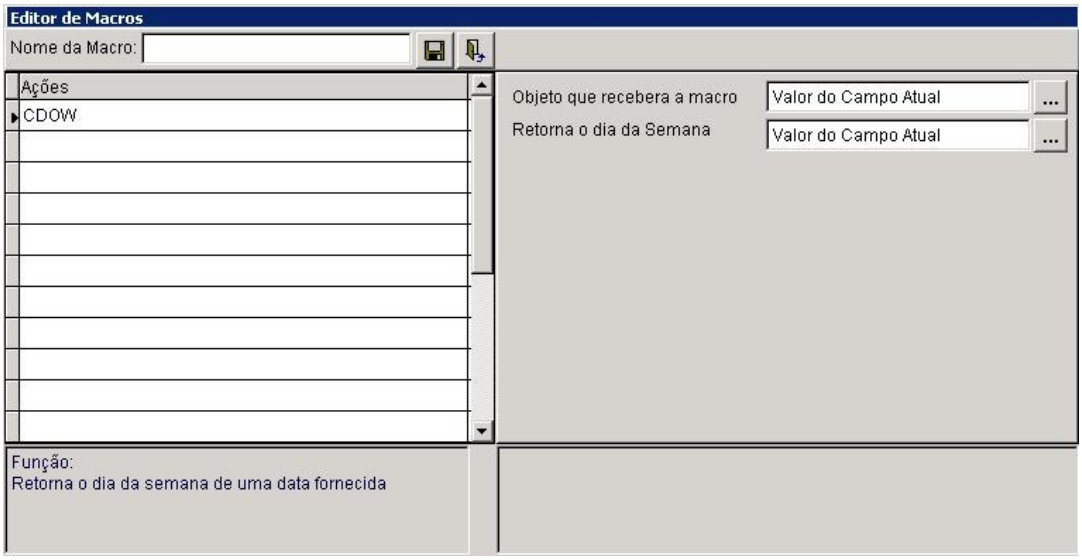


Fig 138 Função *CDOW*

Retorna o número do dia da semana de um campo data

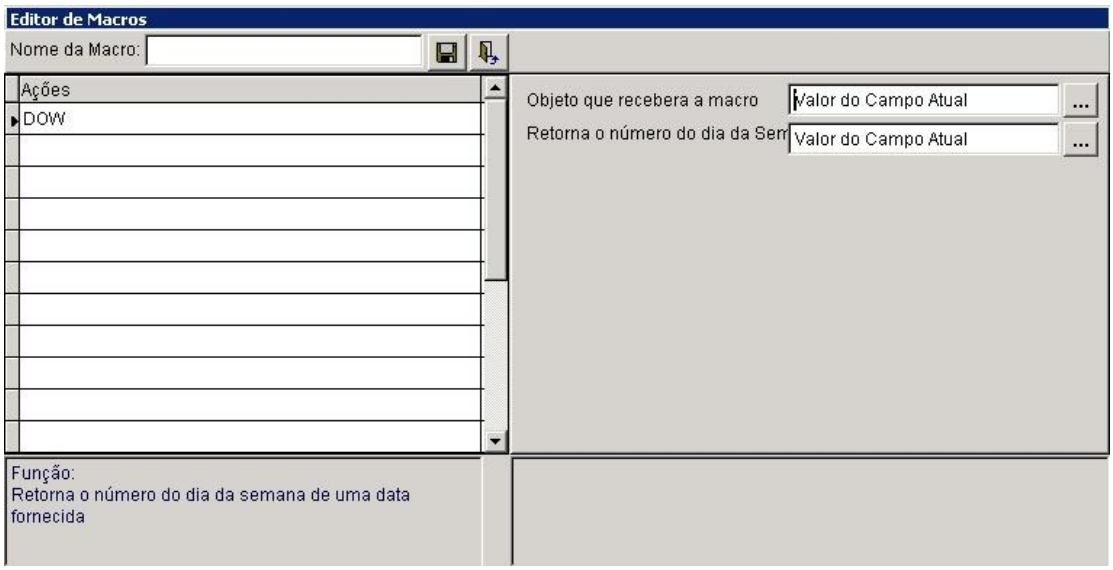


Fig 139 Função *DOW*

Valor de retorno: Numérico

Retorna o Valor Absoluto de um campo numérico

Valor de retorno: Numérico

Exemplo
ABS(-45) Retorna: 45

O Editor de Códigos.

O Visual Kit 5 *Professional* possui o recurso de processador de textos para a digitação de seus códigos.

Para cada evento na janela de propriedades, estará disponível a janela Editor de Códigos, onde você digitará seus código em linguagem Visual FoxPro e no final depurará o código sinalando as linhas com erro.

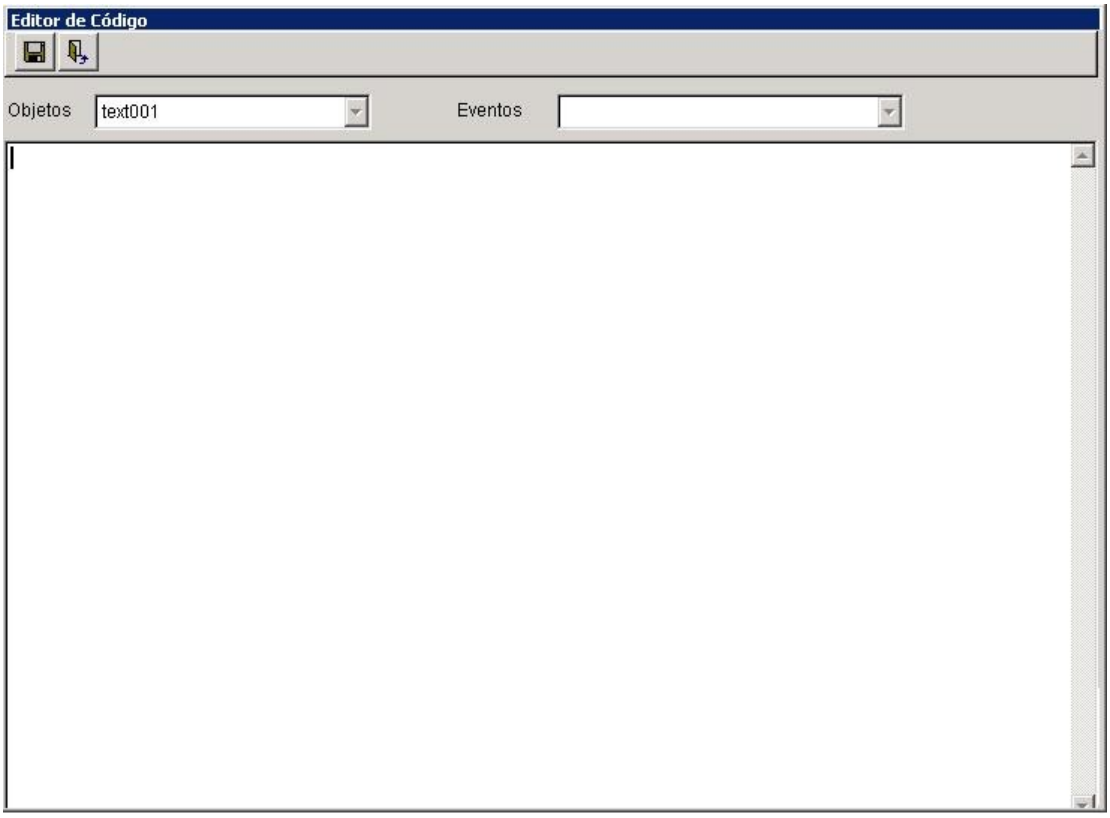


Fig 140 O Editor de códigos

Utilizando o menu *Edit* para auxiliar a digitação de códigos.

Além dos recursos já mencionados, a janela de códigos permite todo o trabalho de um editor de textos padrão Windows. Desta forma sempre que a janela de Editor de Códigos estiver ativa, vários recursos auxiliares estarão disponíveis no ambiente de desenvolvimentos.

A maioria das opções disponíveis no menu Edit funciona como recursos auxiliares à digitação de códigos. Por meio das opções Recortar, Copiar, Colar e Limpar você pode recortar, copiar colar e apagar, respectivamente, o texto selecionado na janela de Editor de Código. A opção Seleciona Tudo, quando em uma janela de códigos, permite selecionar todo o texto disponível na mesma.

VIII. Geração de códigos fontes dos formulários

O Visual Kit 5 *Professional* possui todos os recursos da versão *Standard* e ainda a possibilidade de gerar o código fonte dos formulários das entidades em Fox Pro e Visual Basic.

O objetivo principal da Edição *Professional* é aumentar a produtividades dos programadores. Utilizando os recursos do Visual Kit 5, o programador vai poder desenhar de maneira fácil os formulários que poderão ser incrementados com as linguagens de programação.

Atenção!
Depois de gerados os códigos dos formulários e o usuário ter incrementado e compilado com os recursos oriundos das respectivas linguagens de programação, futuras modificações devem ser feitas diretamente pelos linguagem de programação e não pelo visual Kit 5.

Gerando códigos em Visual Basic

Respeito a geração dos códigos dos formulários em VB é importante ter em conta o seguinte:

- O código dos formulários em Visual Basic é gerado exclusivamente para os formatos de tabelas **Access** e **SQL-Server**. Por tanto a entrada de dados **Driver** da janela de “Dados do Sistemas” deve estar definida para um destes drivers.

- Para compilar o código gerado dos formulários é necessário ter instalado o compilador Visual Basic no seu computador. E preciso indicar o caminho onde se encontra o compilador Visual Basic, este caminho é definido na janela Dados do Sistemas na entrada de dados **Diretório do Visual Basic**. (veja na figura abaixo)

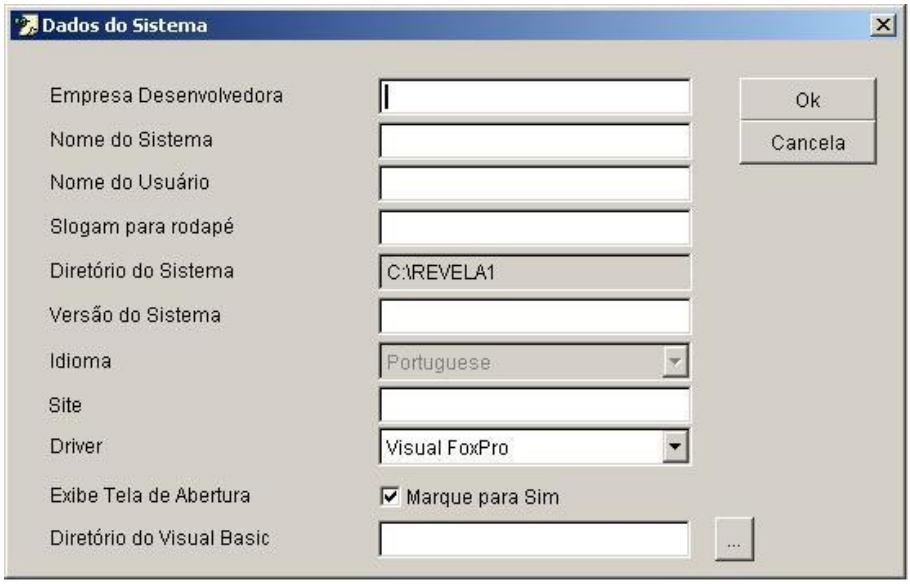


Fig 141 Visual da janela Dados do Sistema da Edição Professional.

Atenção!

Caso não tenha instalado o Visual Basic os códigos serão gerados sem compilar.

- O código será gerado para os formulários das entidades definidas como **Formulário Externo em Visual Basic** na janela Entidades como mostra a seguinte figura.

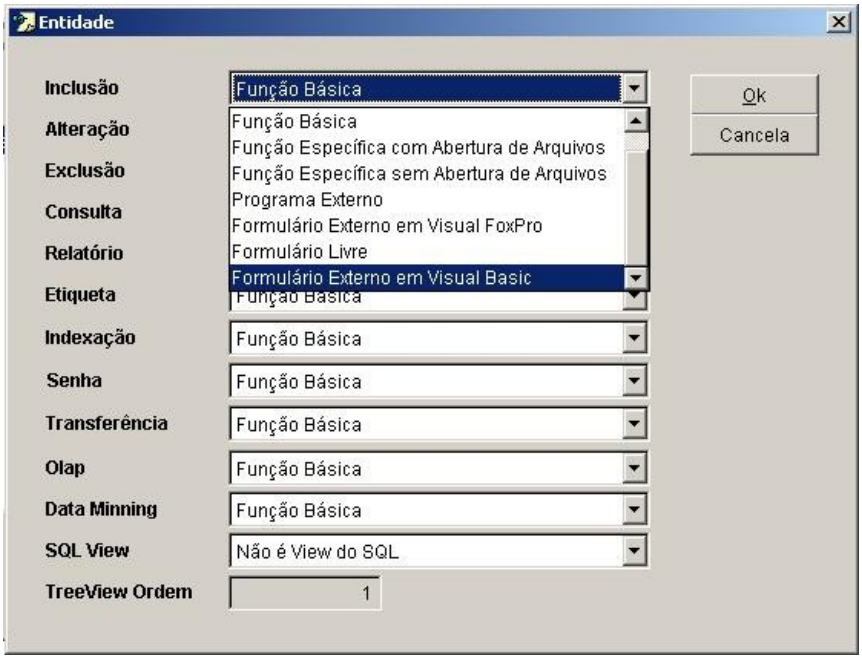


Fig 142 Definindo entidade para geração de código fonte em VB

Depois de desenhado o formulário da entidade, o Visual Kit 5 gera o código em diretórios específicos. Todos os formulários são gravados no sub-diretório do próprio sistema gerado em *PARAMETERS\FORM\D01, D02... D99* de acordo com o módulo ao qual pertence a entidade, no diretório do sistema gerado.

O formulário é gerado em formato .SCX com nome corresponde com a seguinte regra: Exemplo: **F010022501.SCX**

Colunas	Descrição
F	De Formulário
01	Especifica o número do módulo (01) a que pertence
002	Especifica a entidade (002).
2	Com abertura de arquivo. Repete o byte usado nesta coluna no nome da função específica Primitiva.
5	Relatório. Repete o byte usado nesta coluna no nome da função específica Primitiva.
01	Formulário de número 01.

Gerando códigos em Fox Pro

A diferença da geração em Visual Basic, o código em Fox Pro é gerado para todos os formatos disponíveis no Visual Kit 5. E para realizar a compilação não é necessário ter instalado o compilado Fox Pro.

O código será gerado para os formulários das entidades definidas como **Formulário Externo em Visual Fox Pro** na janela Entidades como mostra a seguinte figura.

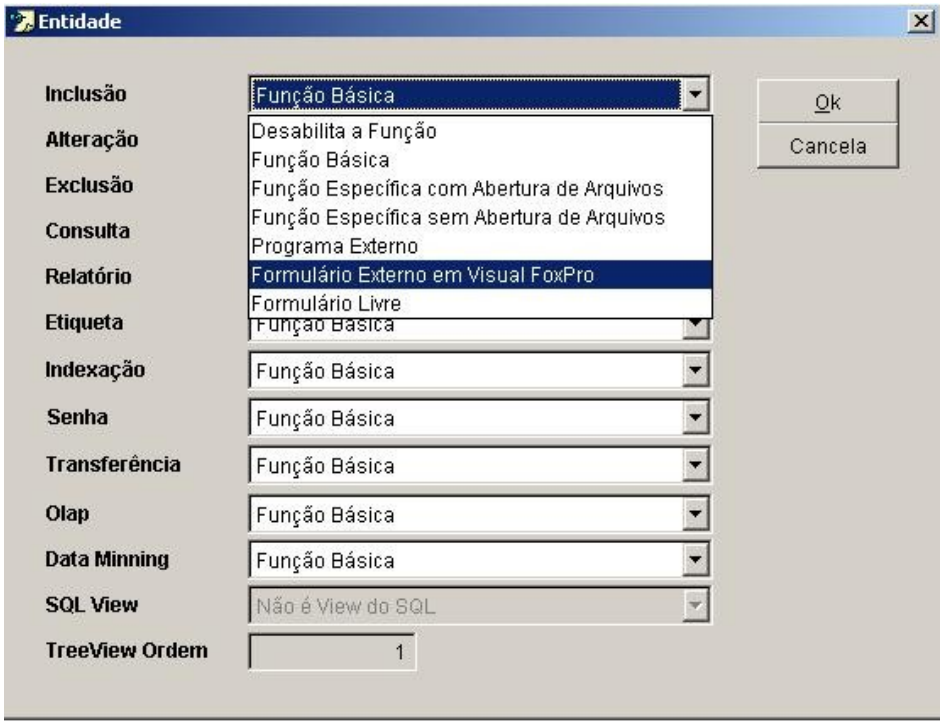


Fig 143 Definindo entidade para geração de código fonte em FoxPro

Depois de desenhado o formulário da entidade, o Visual Kit 5 gera o código em diretórios específicos. Todos os formulários são gravados no sub-diretório do próprio sistema gerado em *PARAMETERS\FORM\D01, D02... D99* de acordo com o módulo ao qual pertence a entidade, no diretório do sistema gerado.

O formulário é gerado em formato .SCX com nome corresponde com a seguinte regra: Exemplo: **F010022501.SCX**

Colunas	Descrição
F	De Formulário
01	Especifica o número do módulo (01) a que pertence
002	Especifica a entidade (002).
2	Com abertura de arquivo. Repete o byte usado nesta coluna

	no nome da função específica Primitiva.
5	Relatório. Repete o byte usado nesta coluna no nome da função específica Primitiva.
01	Formulário de número 01.

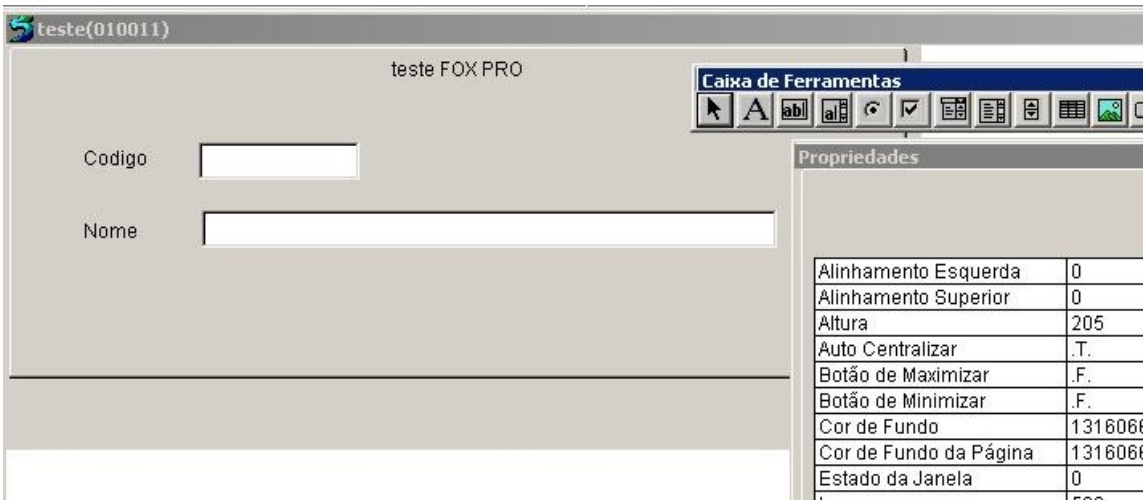


Fig 144 Formulário gerado pelo Visual Kit 5

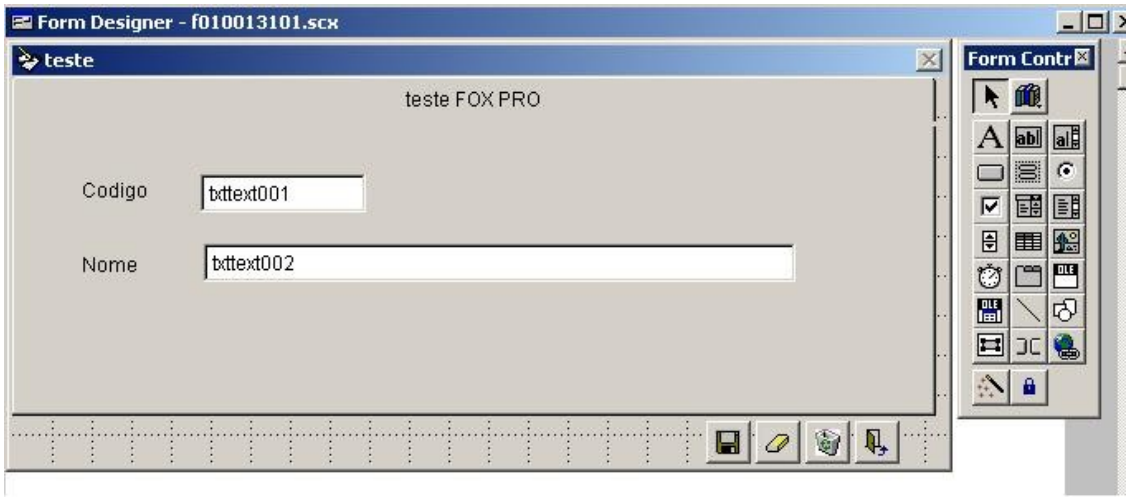


Fig 145 Formulário aberto pelo Visual Fox Pro

IX. Personalizando à apertura, o *about* e o manual

Aqui abordaremos certos procedimentos que darão um toque pessoal a os sistemas gerados pelo Visual Kit 5.

Quando é gerado um sistema pelo Visual Kit 5, são gerados alem dos arquivos relacionados aos formulários e tabelas, uma serie de arquivos complementares. Modificando determinados arquivos complementares você poderá: colocar uma imagem própria na abertura do sistema gerado; colocar uma imagem logo no *About* e colocar um manual específico do sistema criado

Imagem de apertura

Já vimos que, quando começamos o desenvolvimento de um sistema precisamos definir certos dados gerais na janela **Dados do Sistema**.

O último dado desta janela é o **Exibe Tela de Abertura** (veja na Fig 146). Se você marca para sim, o sistema gerado quando aberto mostrará automaticamente (por poucos segundos) a mesma imagem de abertura do próprio Visual Kit 5.

Fig 146 Janela Dados do Sistema

Esta imagem de abertura é gravada no arquivo **show.bmp** na própria pasta do sistema gerado no seguinte caminho: **C:\teste\parameters\bmp** (no exemplo o nome do sistema é "**teste"**)
Você pode substituir o arquivo **show.bmp** por outro com a imagem que desejada, mas o nome deve ser mantido.

O logo do *About*

Automaticamente o sistema é gerado com uma janela de About (Sobre) mostrada na Fig 147, com os dados cadastrados na janela **Datos do Sistema** (veja a figura anterior)

Fig 147 Janela *About*

Você pode incluir uma imagem no quadro branco. Usando o mesmo caminho da seção (**C:\teste\parameters\bmp**) localize o arquivo **logo.bmp**. Abra este arquivo e incluía a imagem desejada ou substituía por outro mas com o mesmo nome.

O manual do sistema gerado

No sistema gerado, quando você clica no botão "**Help"** no menu e escolhe a opção "**Help..."** o sistema abrirá um manual predefinido em formato HTML.

Você pode modificar ou colocar um novo manual específico do sistema que você desenvolveu. Para isso basta substituir na própria pasta do sistema gerado o conteúdo do arquivo **C:\teste\parameters\user manual\html\ default.htm** (no exemplo o nome do sistema é "**teste"**)

X. Criando um Sistema Completo pela *Designer-Básica*

Neste capítulo desenvolveremos um sistema com características bem reais. Usaremos exclusivamente recursos próprios do Visual Kit 5 e mostraremos a melhor maneira de desenvolver sistemas.

A Proposta

Um amigo, dono de uma pequena empresa, sabendo que você comprou o Visual Kit 5 quer que você desenvolva um pequeno sistema para ele. Ele precisa de um sistema de gestão empresarial que controle o cadastro de clientes e fornecedores, o estoque de materiais, o movimento de estoque; que gerencie as contas a pagar e receber - luz, água, telefone, etc - e o movimento de todas as contas de correntes bancárias.

Ele quer que o sistema fique dividido em dois subsistemas: um gerenciado pelo funcionário-estoquista e o outro - relativo às contas - pela sua secretária. E que ele possa acessar os dois plenamente.

Analisando o Sistema

Primeiramente marque uma reunião com seu amigo para obter maiores informações e analisar como é organizado o fluxo da empresa . Ele explica que:

O funcionário-estoquista mantém quatro arquivos diferentes: um arquivo de clientes com os dados pessoais como nome, endereço, sexo, etc.; outro dos fornecedores de materiais com os dados de endereço, contato, telefone, etc. Este funcionário controla também um arquivo de Estoque de Materiais onde tem cadastrado todos os materiais com os dados como: fornecedor do produto, preço unitário, tamanho, etc. (em muitos casos um único fornecedor oferece vários materiais). E por último o arquivo de

entrada e saída dos materiais do estoque, ou seja, o movimento do estoque como produto, data, hora, quantidade, etc.

Veja a Fig 148, aqui representamos esquematicamente os quatro arquivos e seus relacionamentos. O arquivo Estoque que armazena cada material está relacionado com o arquivo de Fornecedores onde um fornecedor pode oferecer vários produtos (relacionamento “um-para-muitos” – veja no Capítulo II), portanto, aparece em vários registros do arquivo Estoque, na figura indicado em vermelho. Por outro lado o arquivo Movimento de Estoque está relacionado, também “um-para-muitos” com o arquivo de Estoque, já que um determinado produto tem várias entradas e saídas (assinalado em azul).

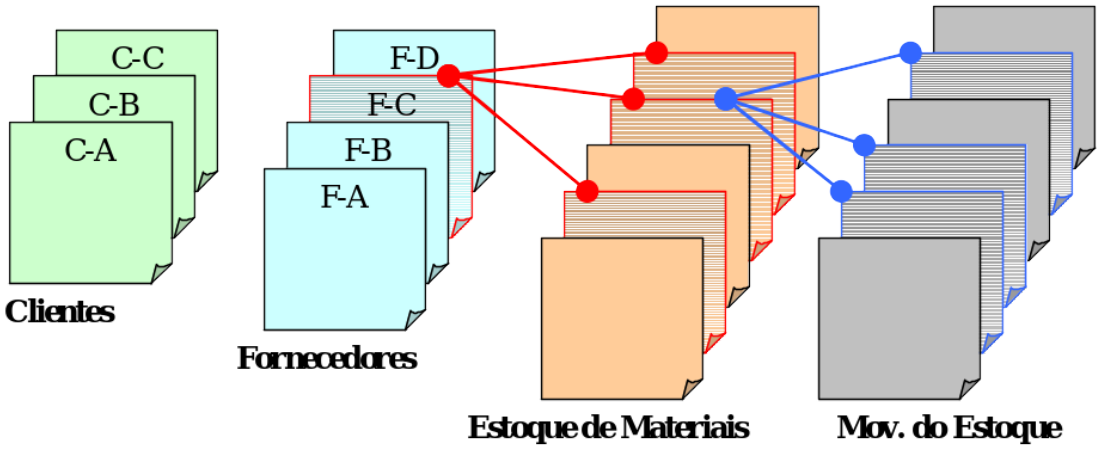


Fig 148 Estoquista

Por sua vez, a secretária leva uma lista com todos os tipos de contas que a empresa trabalha, e um arquivo onde registra os Pagamentos / Recebimentos para cada tipo de conta. Para tanto, existe um relacionamento “um-para-muitos” entre estas duas entidades, onde por exemplo a conta da luz se paga mensalmente (veja na Fig 149 o relacionamento indicado em vermelho).

Também ela guarda uma arquivo com a lista de bancos que a empresa se relaciona com dados como: nome do banco, endereço, gerente, telefone, etc.. E controla os movimentos da conta corrente de cada banco – relacionamento um para muitos-, colocando valor do débito ou crédito, a data e hora, etc.

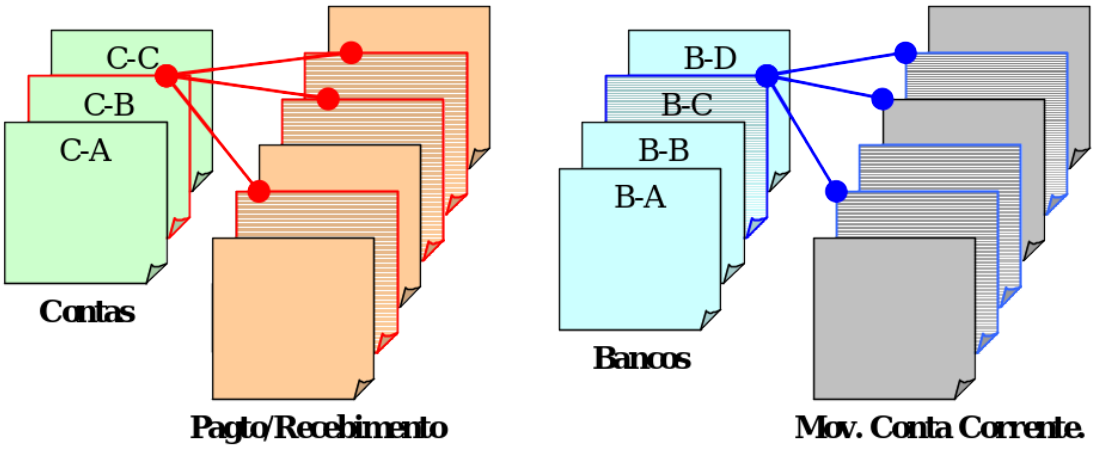


Fig 149 Secretaria

Projetando este panorama para as definições estudadas do Visual Kit 5, a estrutura do sistema poderia estar composta por dois módulos englobando entidades que tenham a ver uma com outras, por exemplo:

MÓDULO	ENTIDADES
01: Controle de Estoque	001 - Clientes 002 - Fornecedores 003 - Estoque de Materiais (relacionado com Fornecedores) 004 - Movimento do Estoque (relacionado com Estoque)
02: Contas a Pagar / Receber	001 - Contas 002 - Pagamentos / Recebimentos (relacionado com Contas) 003 - Bancos 004 - Movimento de Conta Corrente (relacionado com Bancos)

É considerada uma boa prática de desenvolvimento de sistemas, definir - antes de começar desenhar propriamente os formulários-, a estrutura dos mesmos, ou seja, definir quantos campos vai ter o formulário; o tipo de dados que vai guardar, se caractere ou numérico, etc.; o tamanho máximo de caracteres no campo; o tipo de controle na entrada de dados (*TextBox*, *ComboBox)*; os relacionamentos; as validações; etc. Esta fase é chamada de Análise de Sistemas.

Uma maneira simples de fazer esta análise, é colocar em tabelas toda esta informação. É melhor deter-se um pouco na confecção destas tabelas, porque depois será mais fácil e produtivo criar os formulários seguindo um

roteiro definido. Vejamos a continuação, as tabelas com as estrutura para cada entidade:

Estrutura do Módulo 01 - Controle de Estoque

B01001- Clientes

POSIÇÃO	NOME	TIPO	TAMANHO	MÁSCARA	FORMATO	TIPO DE CONTROLE	PÓS VALIDAÇÃO	OBSERV.
1	Codigo	Caractere	5	99999	!X	TextBox	dup_chk(vdados[I],1,"Código",tipo,I,impre)	Código do Cliente (Chave Primária = .T. Verdadeiro.
2	Nome	Caractere	45	XXX... (45)	!X	TextBox		
3	Sexo	Numérico	2 Decimal:0	-	-	Option; 2 opções		1-Fem; 2-Masc
4	EstCivil	Numérico	2 Decimal:0	-	-	Option; 4 opções		Estado civil Solteiro, Casado,...
5	Data_Nasc	Data	8	##/##/####	D	Spiner (min=0; max=10)		Data de Nascimento
6	Filhos	Numérico	2 Decimal:0	-	-	TextBox		Numero de filhos
7	Endereço	Caractere	40	XXX... (40)	!X	TextBox		
8	Bairro	Caractere	15	XXX... (15)	!X	TextBox		
9	Cidade	Caractere	25	XXX... (25)	!X	TextBox		
10	UF	Caractere	2	XX	!X	TextBox		Estado
11	CEP	Caractere	8	99999-999	R	TextBox		
12	Telefone	Caractere	14	(9999)9999-9999	R	TextBox		
13	Fax	Caractere	14	(9999)9999-9999	R	TextBox		

14	Valor_Pago	Numérico	10 Decimal:2	9,999,999.99	R	TextBox		
15	carro					CheckBox		
16	casa					CheckBox		
17	plano					CheckBox		
18	cartao					CheckBox		
19	Descriçã o	Memo	4			EditBox		Edição de outras Descrições dos clientes

Antes de começar a desenhar o sistema, você precisa também ter uma idéia aproximada do visual dos formulários. Adiantamos aqui as telas finais dos formulários, para que o ajudem a desenvolver este sistema.

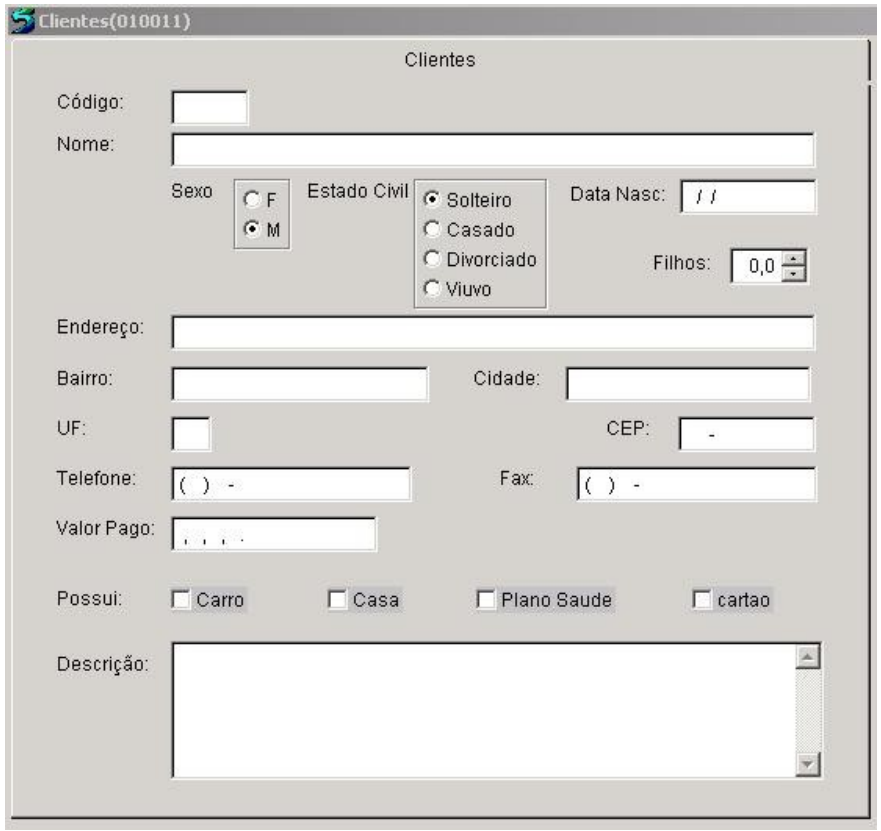


Fig 150 Formulário da entidade Clientes

B01002 - Fornecedores

POSIÇÃO	NOME DO CAMPO	TIPO	TAMANHO	MÁSCARA	FORMATO	TIPO DE CONTROLE	PÓS VALIDAÇÃO	OBSERV.
1	Cód_Fornec	Caractere	5	99999	!X	TextBox	dup_chk(vdados[I],1,"Código",tipo,I,impre)	Código do Fornecedor (Chave P).
2	Nome	Caractere	45	XXX...(45)	!X	TextBox		
3	Endereço	Caractere	40	XXX...(40)	!X	TextBox		
4	Bairro	Caractere	15	XXX...(15)	!X	TextBox		
5	Cidade	Caractere	25	XXX...(25)	!X	TextBox		
6	UF	Caractere	2	XX	!X	TextBox		Estado
7	CEP	Caractere	8	99999-999	R	TextBox		
8	Contato	Caractere	20	XXX...(20)	!X	TextBox		
9	Telefone	Caractere	14	(9999)9999-9999		TextBox		
10	Fax	Caractere	14	(9999)9999-9999		TextBox		
11	Observaçõ es	Memo				EditBox		Edição de observações Ficará em nova pagina.

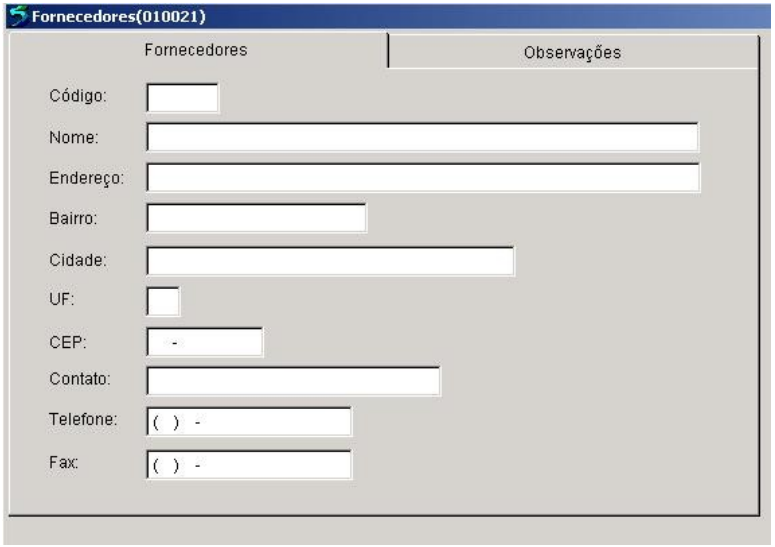


Fig 151

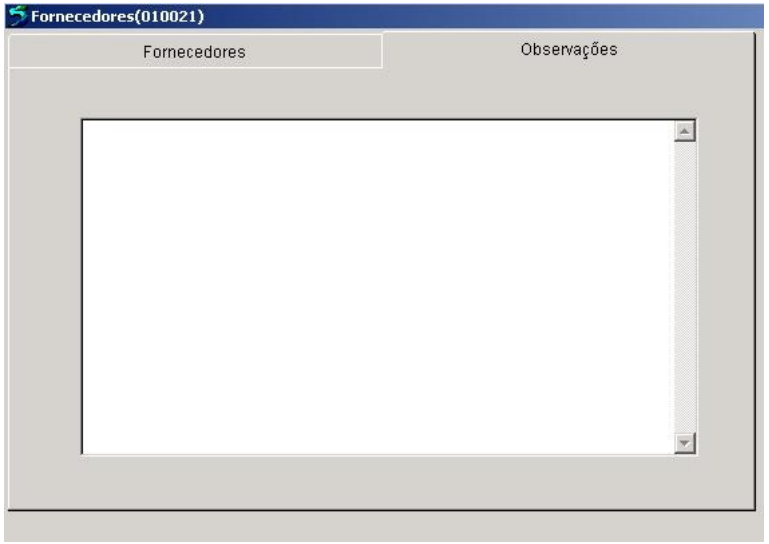


Fig 152

B01003 - Estoque de Materiais

POSIÇÃO	NOME DO CAMPO	TIPO	TAMANHO	DECIMAIS	MÁSCARA	FORMATO	TIPO DE CONTROLE	PÓS VALIDAÇÃO	OBSERV.
1	Cod_Materi	Caractere	5		99999	K	TextBox	dup_chk(vdados[I],1,"Material",tipo,I,impre)	Código do Material (Chave)
2	Nome	Caractere	45		XXX...(45)	!X	TextBox		
3	Cod_Fornec	Caractere	5				ComboBox	dup_chk(vdados[I],2,"Fornecedor",.f.,I,impre)	Cód Fornecedor relacionado com Fornecedores (01002)
4	Cor	Caractere	10		XXX...(10)	!X	TextBox		
5	Tamanho	Numérico	10		9,999,999.99	R	TextBox		
6	Estoq_Mini	Numérico	6		999.99	K	TextBox		Estoque Mínimo
7	Quantidade	Numérico	6		999.99	K	TextBox	iif(vdados[I]>=vdados[I-1],.T.,.F.)	Não pode ser menor que estoque mínimo
8	Preco_Unit	Numérico	12	2	999,999999.99	R	TextBox		Preço Unitário
9	Preco_Ve	Numéri	12	2	999,999999	R	TextBox		Preço de

	nd	co			.99				Venda

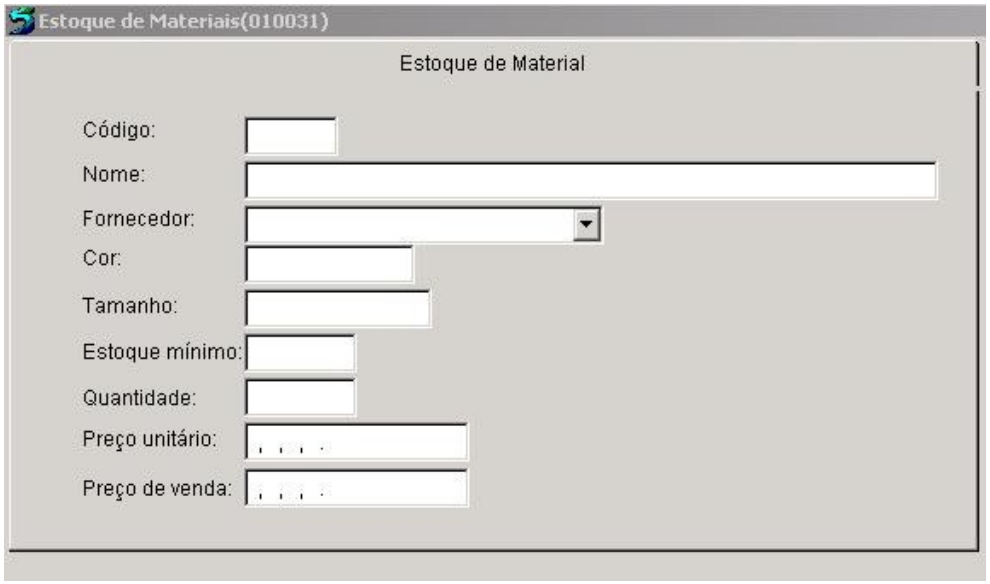


Fig 153 Formulário da entidade Estoque de Materiais.

B01004- Movimento do Estoque

POS	NOME DO CAMPO	TIPO	TAMANHO	MÁSCARA	FORMAT	Controle	Pré validação	Pós validação	OBSERV.
1	Cod_Materi	Caractere	5	99999	K	ComboBox		dup_chk(vdados[I],1,"Código",tipo,I,impre) .and. dup_chk(vdados[I],2," Código ",.f.,I,impre)	Cód do Material (Chave) relacionado com arquivo de Estoque (01003)
2	Data	Data	8	##/##/####	D	TextBox			Data Movimento (Chave)
3	Hora	Caractere	5	99:99	R	TextBox			Hora Movimento (Chave)
4	Sai_Entra	Caractere	1	X	!X	TextBox		(vdados[I] $ “SE”)	“S” para Saida. “E” para Entrada.
5	Quantidade	Numérico	6	999999	K	TextBox		iff(vdados[I]=“E”,chg_data(2,vdados[1],.t.,“”,7,B->QUANTIDADE+vdados[I]),chg_data(2,vdados[],.t.,“”,7,B->QUANTIDADE-vdados[I]))	
6	Preco_Vend	Numérico	12 Decimal:2	999,999,999.99	R	TextBox			Preço de Venda
7	Total	Numérico	12 Decimal:2	999,999,999.99	R	TextBox	altval(vdados,I,vdados[I-1]*vdados[I-2])		Quantidade * Preço de Venda

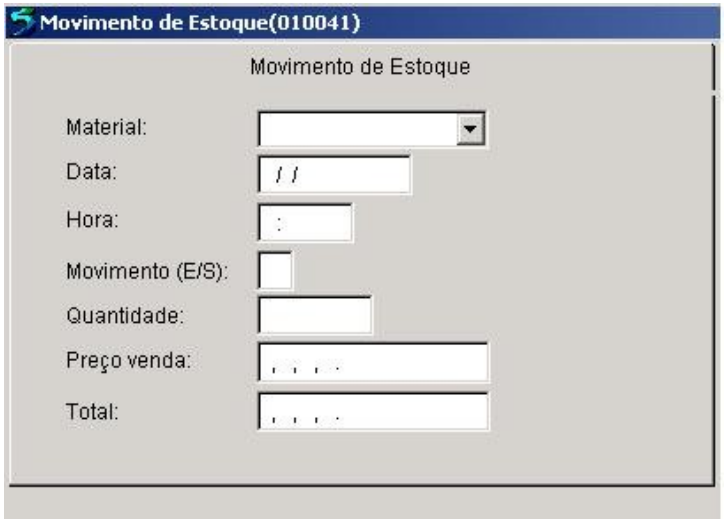


Fig 154 Formulário da entidade Movimento de Estoque.

Estrutura do Módulo 02 – Contas a Pagar/Receber

B02001 - Contas

POSIÇÃO	NOME DO CAMPO	TIPO	TAMANHO	MÁSCARA	FORMATO	Tipo de Controle	Pós validação	OBSERV.
1	Cod_Conta	Caractere	5	99999	K	TextBox	seq_().and.dup_chk(vdados[I],1,"COD_CONTA",tipo,I,impre)	Código da Conta (Chave). Inclusão seqüencial
2	Nome	Caractere	45	XXX...(45)	!X	TextBox		

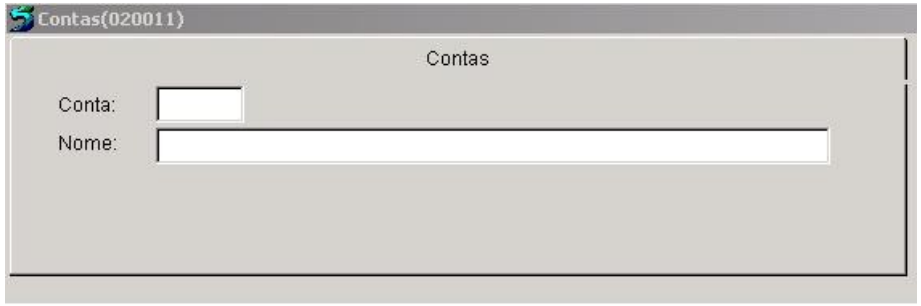


Fig 155 Formulário Contas

B02002 - Pagamentos/Recebimentos

POSIÇÃO	NOME DO CAMPO	TIPO	TAMANHO	DECIMAIS	MÁSCARA	Tipo de Controle	Pré validação	Pós validação	OBSERV.
1	Cod_Conta	Caractere	5			ComboBox			Código da Conta (Chave) relacionado com arquivo de Contas (02001)
2	Data	Data	8		##/##/####	TextBox			Data Pag/Rec (Chave)
3	Hora	Caractere	5		99:99	TextBox			Hora Pag/Rec (Chave)
4	Pagou_Rec	Caractere	1		A	TextBox		(vdados[I] $ “PR”)	“P” para pagamento. “R” para Recebimento.
5	Valor	Numérico	12	2	9,999,999,999.99	TextBox			Valor Pago / Recebido
6	Multa	Numérico	12	2	9,999,999,999.99	TextBox			Valor da Multa
7	Total	Numérico	12	2	9,999,999,999.99	TextBox	altval(vdados,I,vdados[I-1]+vdados[I-2])		Valor + Multa

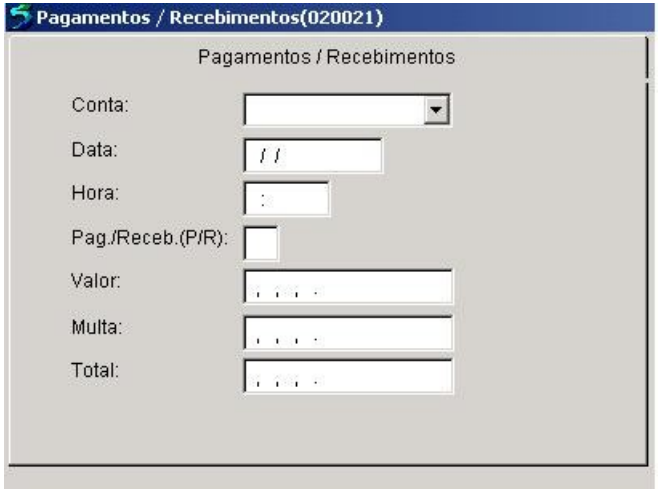

Pagamentos / Recebimentos(020021)
Pagamentos / Recebimentos
Conta:
Data:
/ /
Hora:
:
Pag./Receb.(P/R):
Valor:
Multa:
Total:

Fig 156

B02003 – Bancos

POSIÇÃO	NOME DO CAMPO	TIPO	TAMANHO	MÁSCARA	TIPO DE CONTROLE	PÓS VALIDAÇÃO	OBSERV.
1	Num_Banco	Caractere	5		TextBox	seq_().and.dup_chk(vdados[I],1,"Num_Banco",tipo,I,impre)	chave
2	Nome	Caractere	45		TextBox		
3	Endereço	Caractere	40		TextBox		
4	Bairro	Caractere	15		TextBox		
5	Cidade	Caractere	20		TextBox		
6	UF	Caractere	2		TextBox		
7	CEP	Caractere	8	99999-999	TextBox		
8	Gerente	Caractere	20		TextBox		
9	Telefone	Caractere	14	(9999)9999-9999	TextBox		
10	Fax	Caractere	14	(9999)9999-9999	TextBox		

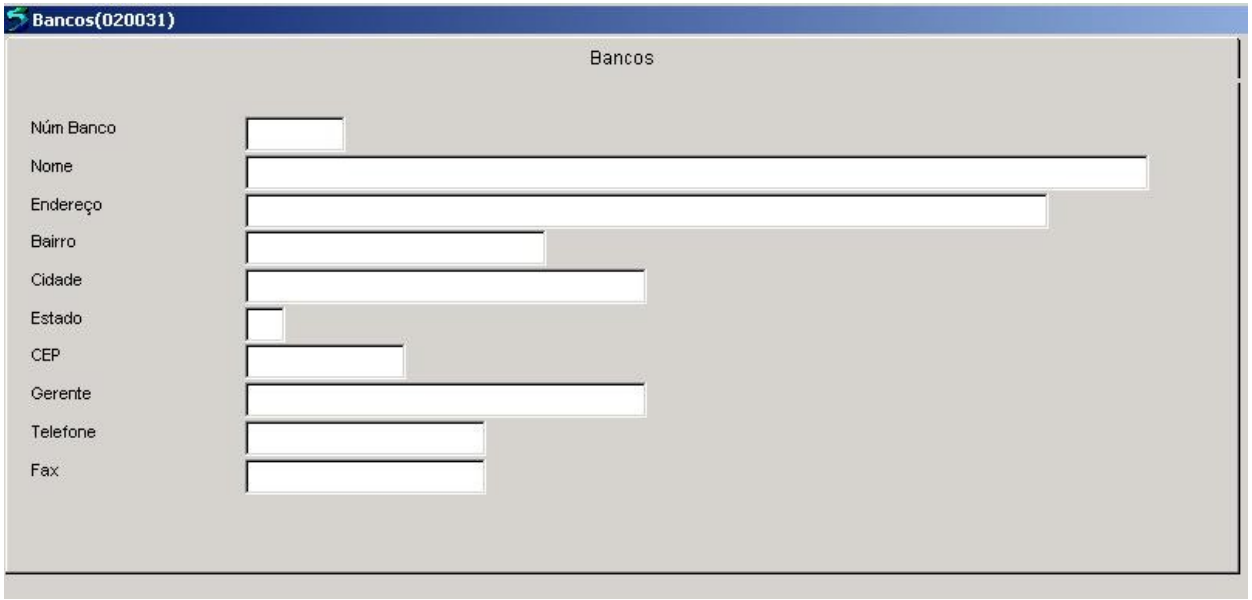

Bancos(020031)
Bancos
Núm Banco
Nome
Endereço
Bairro
Cidade
Estado
CEP
Gerente
Telefone
Fax

Fig 157

2004 - Movimento de Conta Corrente

POSIÇÃO	NOME DO CAMPO	TIPO	TAMANHO	DECIMAIS	MÁSCARA	Tipo de Controle	Pós validação	OBSERV.
1	Num_Banco	Caractere	5			TextBox		Número do Banco (chave). Relacionamento: 02003
2	Conta	Caractere	12		999999999999	TextBox		Numero de conta corrente. (chave)
3	Num_cheque	Caractere	12		999999999999	TextBox		
4	Data	Data	8		##/##/####	TextBox		
5	Déb_Cred	Caractere	1			TextBox	(vdados[I] $ “DC”)	D para Débito C para Credito
6	Valor	Numérico	12	2	9,999,999,999.99	TextBox		Valor do credito/debito.

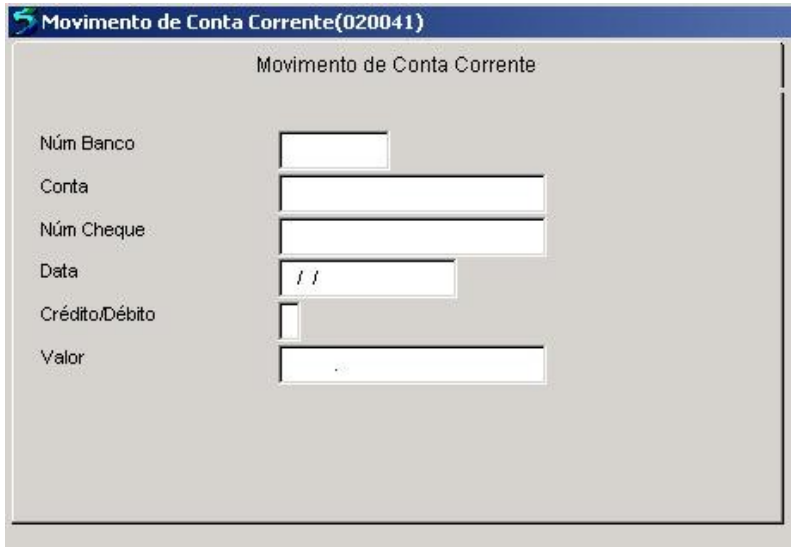


Fig 158 Formulário Movimento de Conta corrente

Passos na Criação do Sistema

Continuando, colocaremos de maneira geral todos os passos para a criação do sistema usando o Visual Kit 5.

- Em primeiro lugar, inicialize o Visual Kit 5
- Abra a pasta “Gerador de Sistema”, clique em “Sistema Completo” e selecione a opção “Inclusão” -> “Sistema Completo”. Gere o diretório para o sistema. **\Controle de Empresa.**
- Preencha os dados gerais do sistema.
- Crie os módulos *Controle de Estoque* e *Contas a Pagar/Receber* como foi explicado no capítulo IV. Veja na figura seguinte.

Fig 159 Criação dos Módulos

- Clique em *Controle de Estoque* e cadastre as entidades. Veja na Fig 160.
- Clique logo em *Contas a Pagar/Receber* e cadastre as entidades correspondentes. Veja na Fig 161.

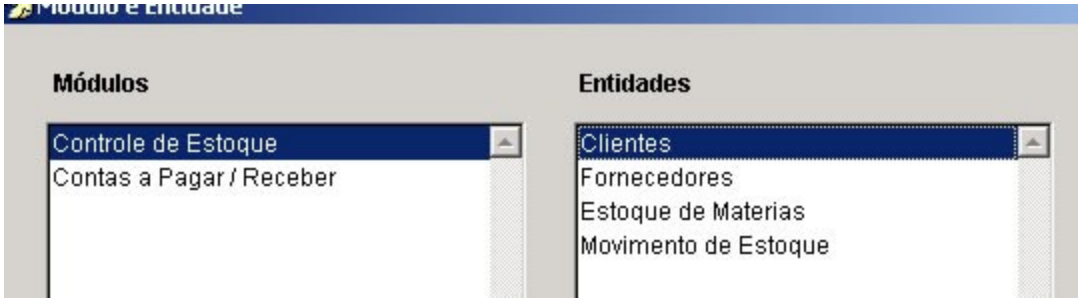


Fig 160 Cadastrando as entidades de *Controle de Estoque*.

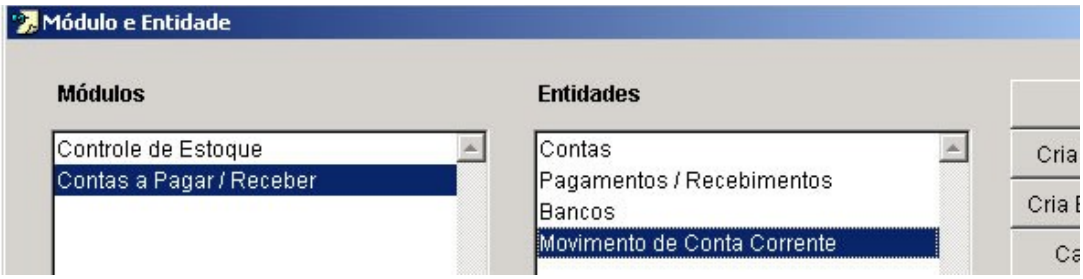


Fig 161 Cadastrando as entidades de *Contas a Pagar/Receber*.

- Quando você cria uma entidade, o Visual Kit 5, solicita a confirmação de quais operações básicas serão herdadas pela Entidade. Selecione as operações.
- Para desenhar o formulário de *Clientes*. Clique no módulo *Controle de Estoque* e posteriormente clique na entidade *Clientes*. Confirme a seleção segundo aparece na Fig 162.

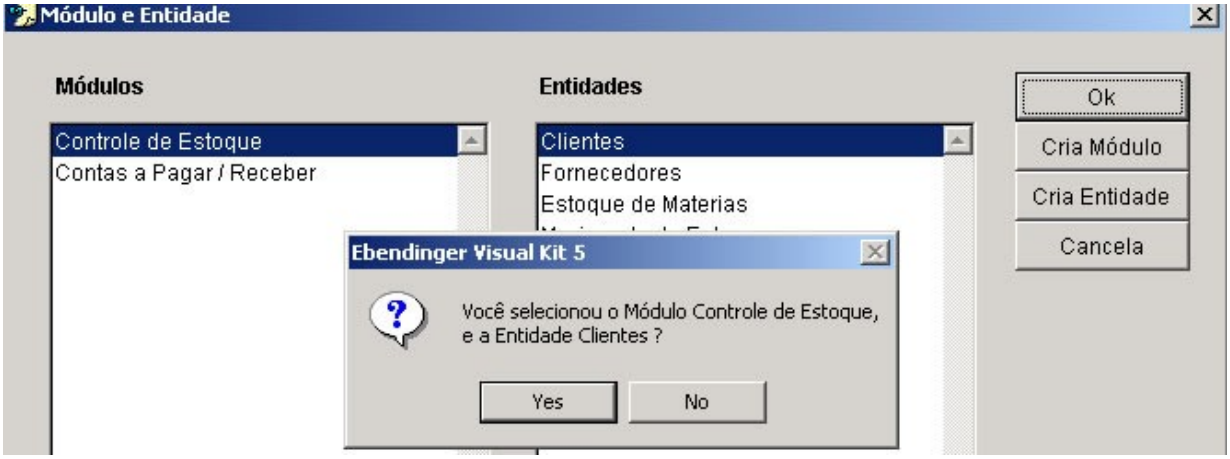


Fig 162 Selecionando a Entidade Clientes.

- Insira os controles no formulário.
- Distribuí-los e dimensioná-los de forma agradável e objetiva.
- Altere os valores de suas propriedades para personalizá-los segundo a tabela de definição da estrutura.
- Faça o mesmo para as outras entidades do sistema.

Veja alguns conselhos importantes durante o desenho dos formulários.:

- Se durante o desenho da tela, um dos campos ocupar uma área maior que a janela, você poderá com o mouse redimensionar o campo. Veja exemplo na figura abaixo:

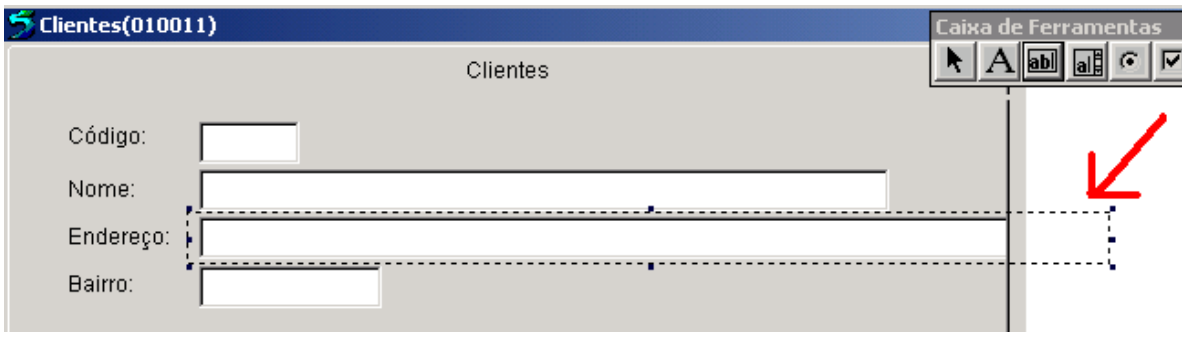


Fig 163 Redimensionando o controle.

- Sempre que o campo for chave, selecione a propriedade "Chave Primária" e altere o valor para .**T**. - Verdadeiro.

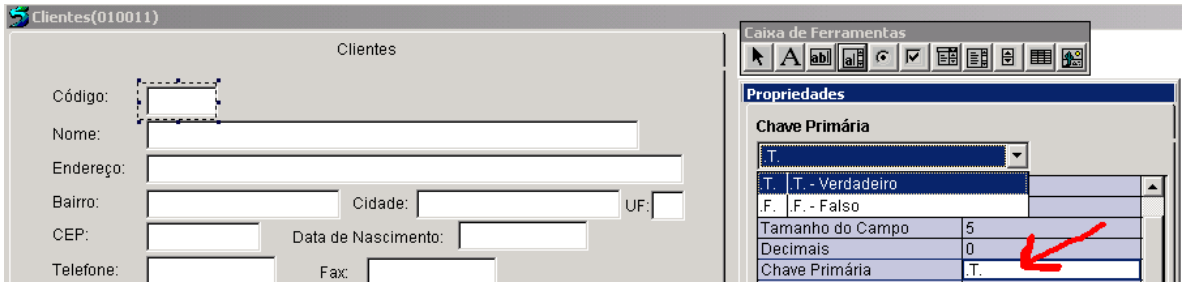


Fig 164 Colocando Chave primaria.

- Sempre que houve relacionamento preencha o campo indicado abaixo, com o número do módulo e da entidade relacionada.

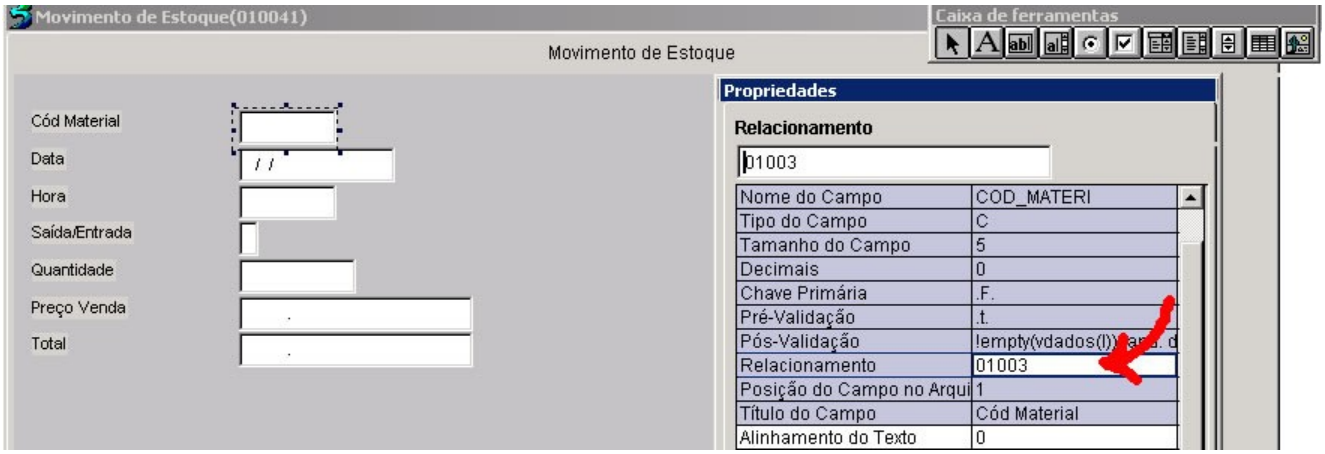


Fig 165 Colocando relacionamento.

- Não esqueça de salvar, pressione periodicamente botão **Salvar**, e no final de cada desenho do formulário;
- Se tudo correu bem, o sistema gerado ficou como mostra a Fig 166.

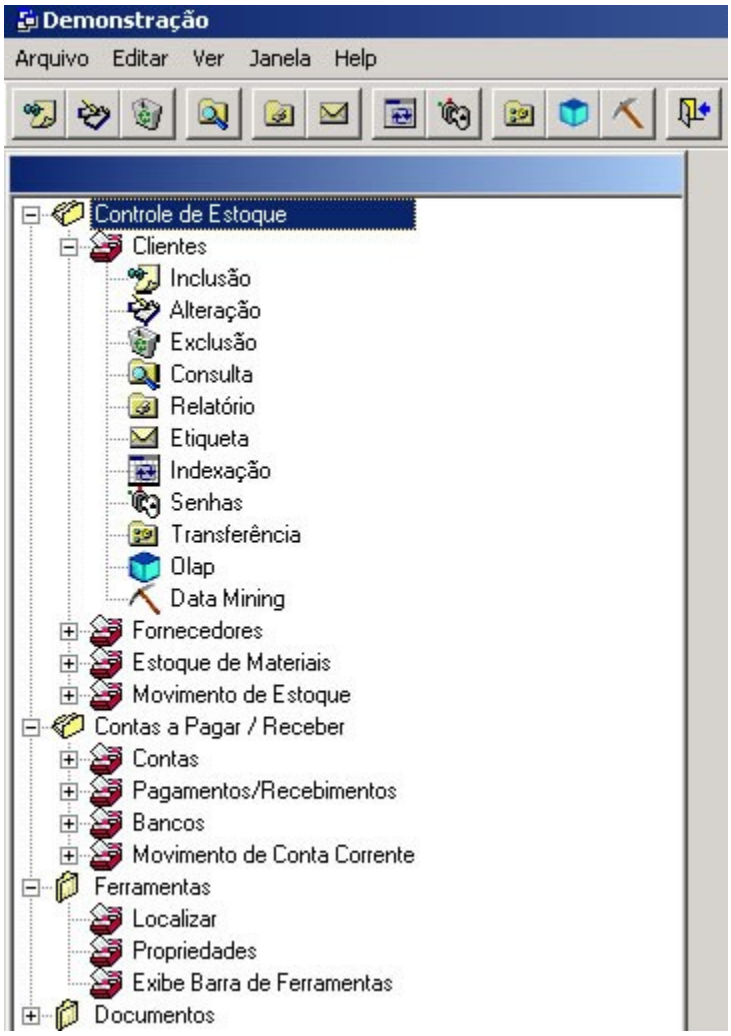


Fig 166 Sistema Gerado

Dica: Gere sistemas brindes e navegue por eles, para entender o funcionamento do Visual Kit 5 e dos sistemas gerados.

XI. Criando Sistemas Utilizando os Sistemas Brinde

Para desenvolver sistemas você não precisa começar de zero. O Visual Kit 5 disponibiliza uma biblioteca de sistemas prontos para funcionar. Esta biblioteca contem dezenas de sistemas que abarcam diversas áreas de negócios, por exemplo: Turismo, Protocolo, Contabilidade, etc.

Os sistemas brinde são um novo conceito de desenvolvimento de sistemas. O Visual Kit 5 cria um sistema completo sem a necessidade de nenhuma interferência,- basta selecionar o sistema da lista -, posteriormente você pode alterar este sistema personalizando-o e comercializando-o sem nenhuma restrição.

Também é possível importar sistemas desenvolvidos com o Visual Kit 5, para fazer parte da biblioteca ou lista de “Sistemas Prontos”. Ou seja a tendência é que com o aumento de sua biblioteca de sistema, aumente sua produtividade e facilidade no desenvolvimento de sistemas.

A lista completa dos sistemas você encontra na pagina da Ebendinger, onde você poderá abaixar para seu computador. Os usuários cadastrados podem baixar gratuitamente sistemas brinde liberados no futuro.

Gerando o sistema brinde

Os procedimentos para criar sistemas baseados nos sistemas brinde é similar ao procedimento de criar sistemas novos visto anteriormente:

- Clique na *TreeView* para abrir a pasta “Gerador de Sistema”,
- Clique em “Sistema Completo” e selecione a opção “Inclusão”.
- É apresentada uma caixa de diálogos intitulada “Utiliza Sistemas Prontos / Cria Sistema Novo” conforme mostra a Fig 167..
- Selecione **Gerar Sistema Brinde**.

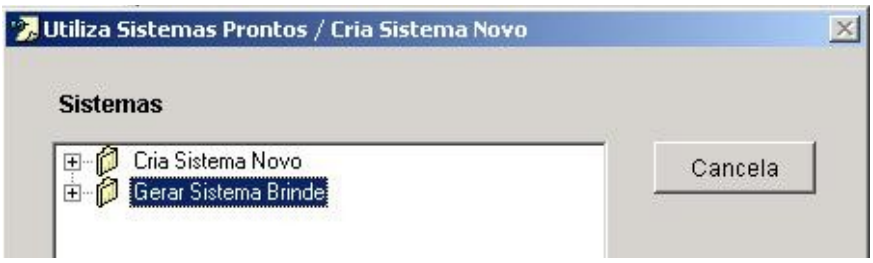


Fig 167 Selecionando Gerar Sistema Brinde

- Aparecerá a lista dos sistemas brinde disponíveis. Utilize a barra de rolagem vertical da janela para ter acesso a todos os sistemas. (Veja na Fig 168).
- Selecione o sistema brinde desejado.

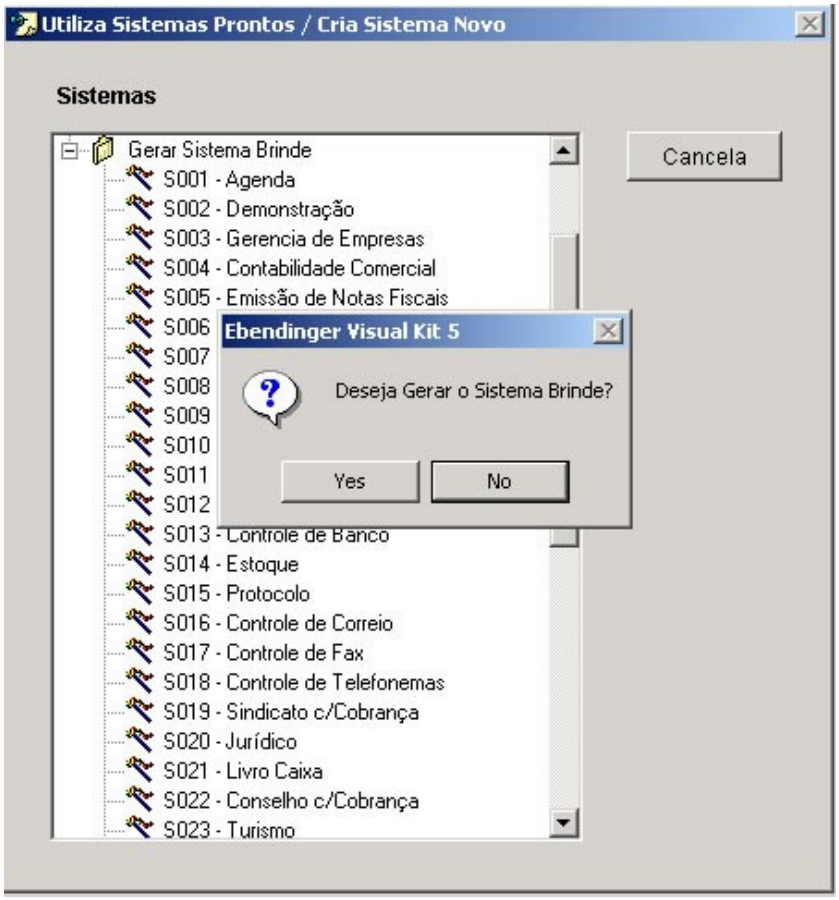


Fig 168 Lista de Sistemas Brinde

- Ao clicar se abrirá uma janela, solicitando que você digite o nome do diretório onde será gerado o sistema. O Visual Kit5 coloca já um nome *default* que pode ser usado diretamente ou mudado.
- No final da criação aparece uma janela informando que o a geração foi concluída (veja na Fig 169)

Fig 169 Sistema Gerado com Sucesso.

Alterando o sistema brinde

Se você deseja fazer alguma modificação ao sistema gerado pela biblioteca, basta abrir o sistema pelo Visual Kit 5, que você terá acesso a todos os formulários. Veja os procedimentos:

- Abra o Visual Kit 5.
- Clique na *TreeView* para abrir a pasta “Gerador de Sistema”
- Clique em “Sistema Completo” e selecione a opção “Inclusão”.
- Na janela digite o caminho do diretório onde está o sistema a modificar ou localize clicando no botão com reticências. (veja na Fig 170). Por *default* nesta janela sempre aparecerá o último sistema editado pelo Visual Kit 5.

Fig 170 Localizando o sistema gerado

- Automaticamente será aberta uma janela como aparece na seguinte figura. Faça um duplo clique no diretorio **c:** ; aparecerão todas as pastas; selecione então a pasta do sistema desejado

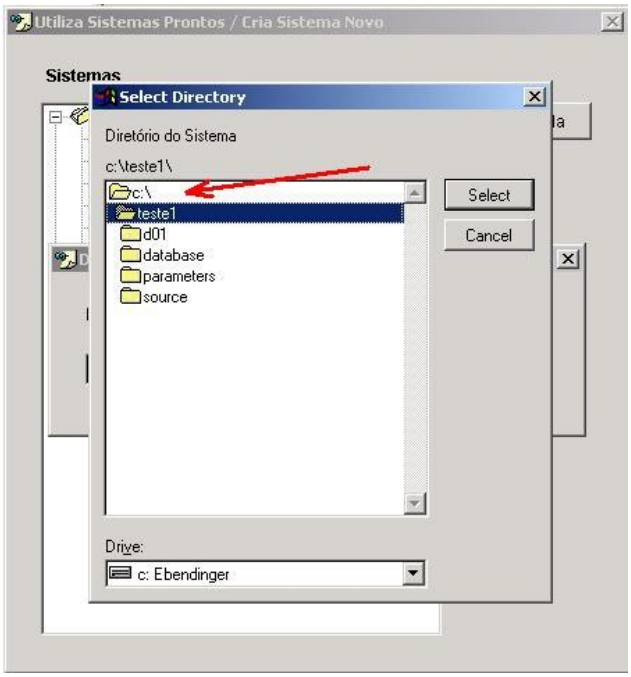


Fig 171 Localizando pasta do sistema

- Confirme. Finalmente será aberta a janela de Módulo e Entidade, permitindo ter acesso para modificar todos os módulos e entidades do sistema.

Atenção

A Ebendinger não oferece suporte aos sistemas brinde, uma vez que eles foram cedidos sem nenhum custo aos usuários do Visual Kit 5.

XII. Aproveitando dados

Muitas vezes já possuímos uma base de dados que desejamos aproveitar para criar um sistema completo. Neste capítulo aprenderemos como fazer isto de forma simples e rápida mostrando os formatos mais usados: DBF, MDB (Access) e XLS (Excel).

Acessando uma tabela .DBF existente

Realize a modo de teste o siguente exemplo:

1) Crie um Banco de Dados, por exemplo: em Fox-Pro e nomeie-o de **ExemploDBF**.
2) Crie uma Tabela e nomeie-a de **B01001**
3) Entre no Visual kit5, clique em **Gerador de Sistemas > Sistema Completo > Inclusão**.
4) Agora clique em **Cria Sistema Novo > Sistema Completo**.
5) Nomeie a pasta do sistema de **c:\SistemaExemplo**.(Ver figura)

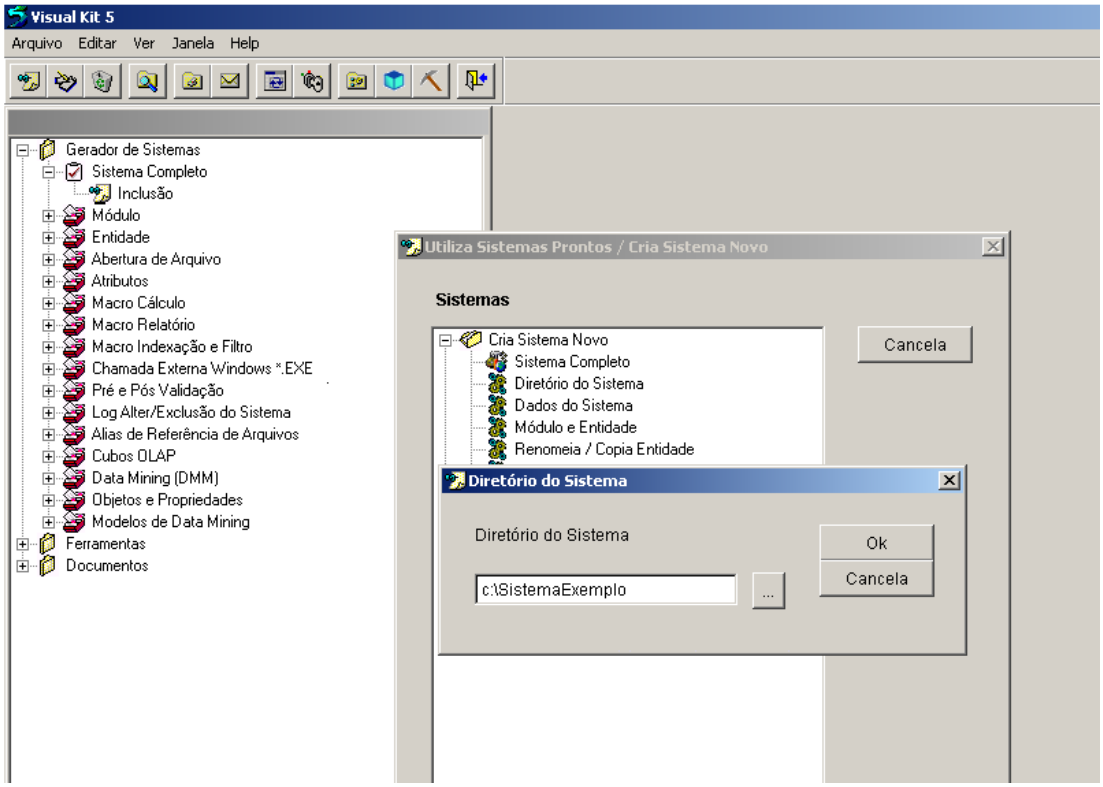


Fig 172

6) Agora crie um **Módulo** e chame-o de **Cadastro**(Ver Fig.02)
7) Crie uma **Entidade** e chame-a de **Usuários**(Ver Figura)

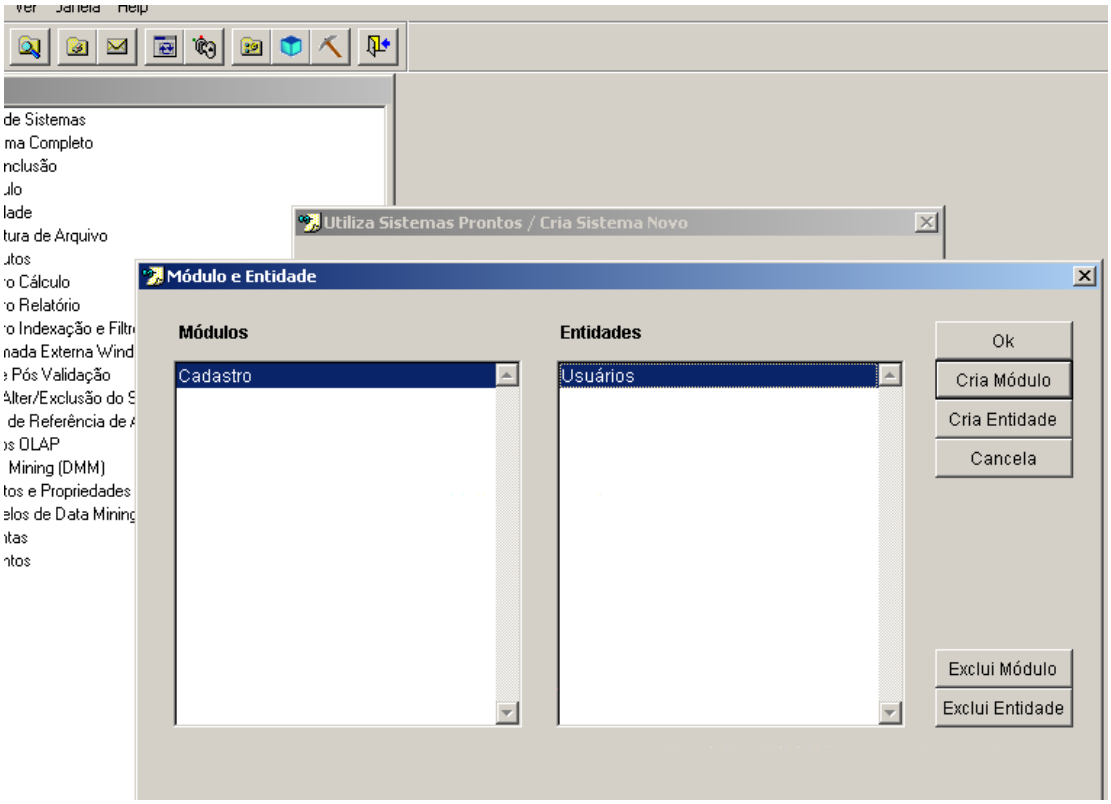


Fig 173

8) Saia do Visual Kit5, entre no diretório **c:** e localize a pasta que contém o Banco de Dados criado no exemplo (**c:\ExemploDBF**)

9) Copie o Banco e cole-o na pasta do sistema criado no caminho: **c:\SistemaExemplo\d01\dbf** (Ver Figura)

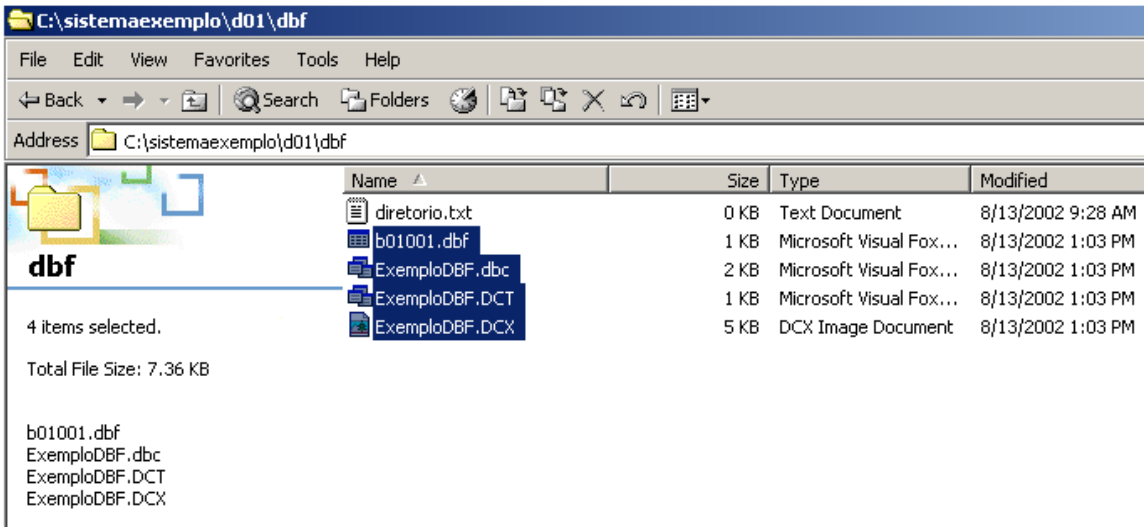


Fig 174

10) Depois de realizar a cópia entre novamente no Visual Kit5
11) Clique em **Abertura de Arquivo > Inclusão** (Ver Figura)

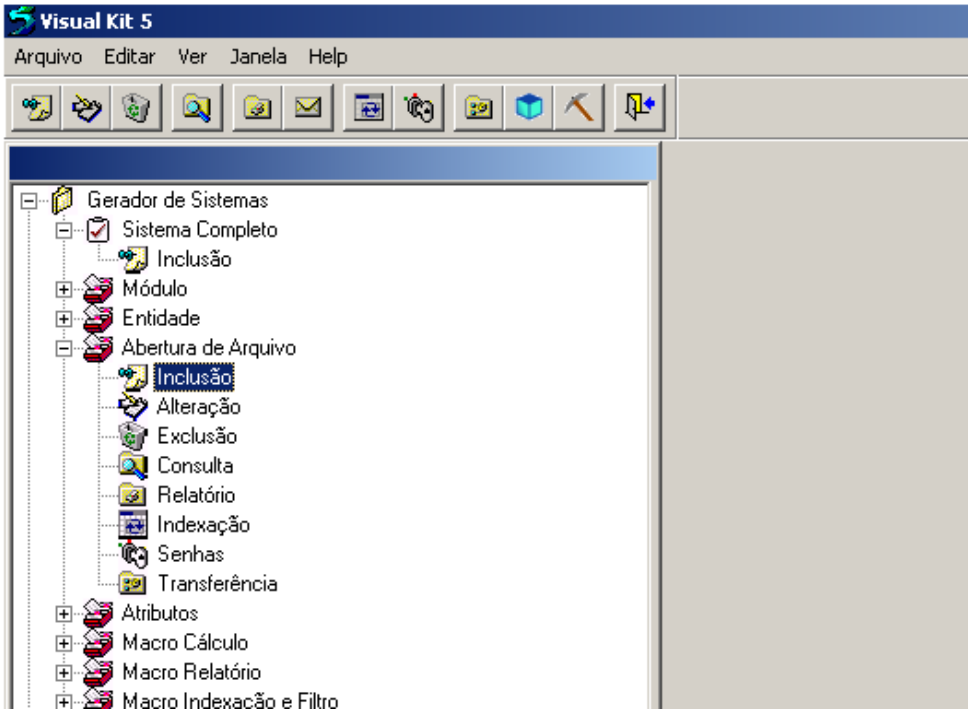


Fig 175

12)Digite os dados desta maneira (ver figura)

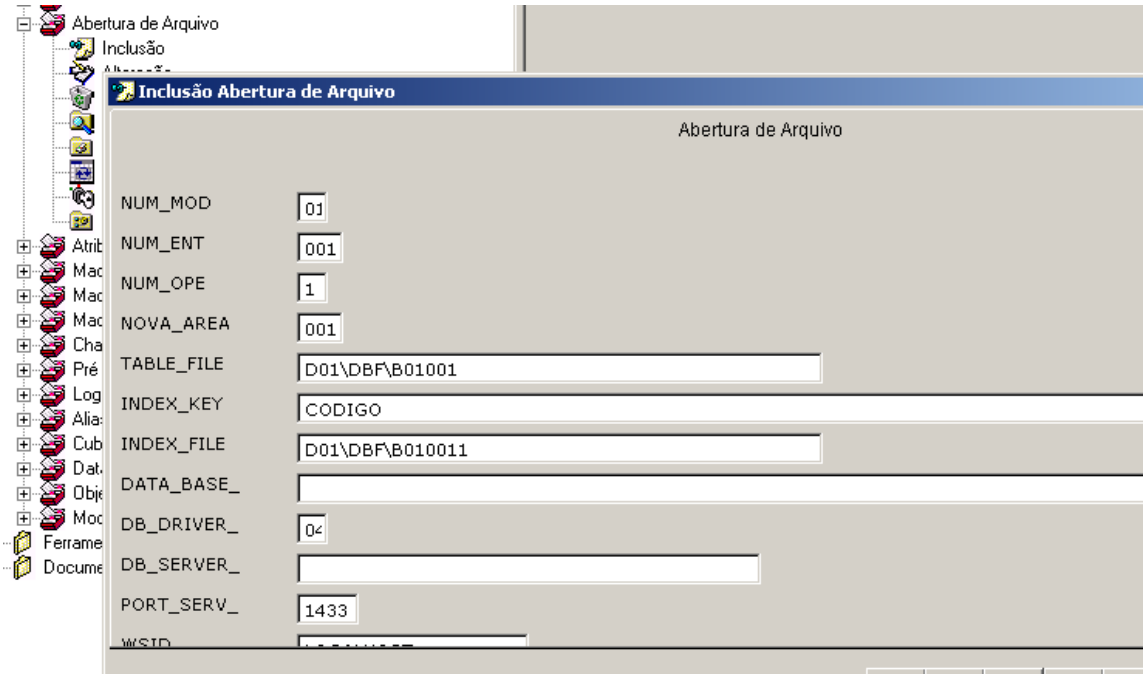


Fig 176

Nº do Módulo= 01
Nº da Entidade= 001
Num_Ope= 1
Nova_Area= 001
Table_File= d01\dbf\B01001
Index_Key= codigo
Index_File= d01\dbf\B010011
Data_Base_= (em branco)
DB_DRIVE_= 04
DB_SERVER_= (em branco)
PORT_SERV_= 1433
WSID= LOCALHOST

13) Agora salve, feche o Visual kit5 e rode o **SistemaExemplo**

Acessando uma tabela .MDB existente

Siga os seguintes passos:

- Crie um Banco de Dados **Acess** e salve-o como "**mdbbase.mdb**".
- Crie uma Tabela e nomeie-a de "**Clientes**".
- Crie pelo menos 2 campos na tabela e inclua alguns registros.

1) Entre no Visual Kit5, crie um Sistema e chame-o de "**Exemplo_Acess**" e em seguida clique em **OK** (ver Figura)

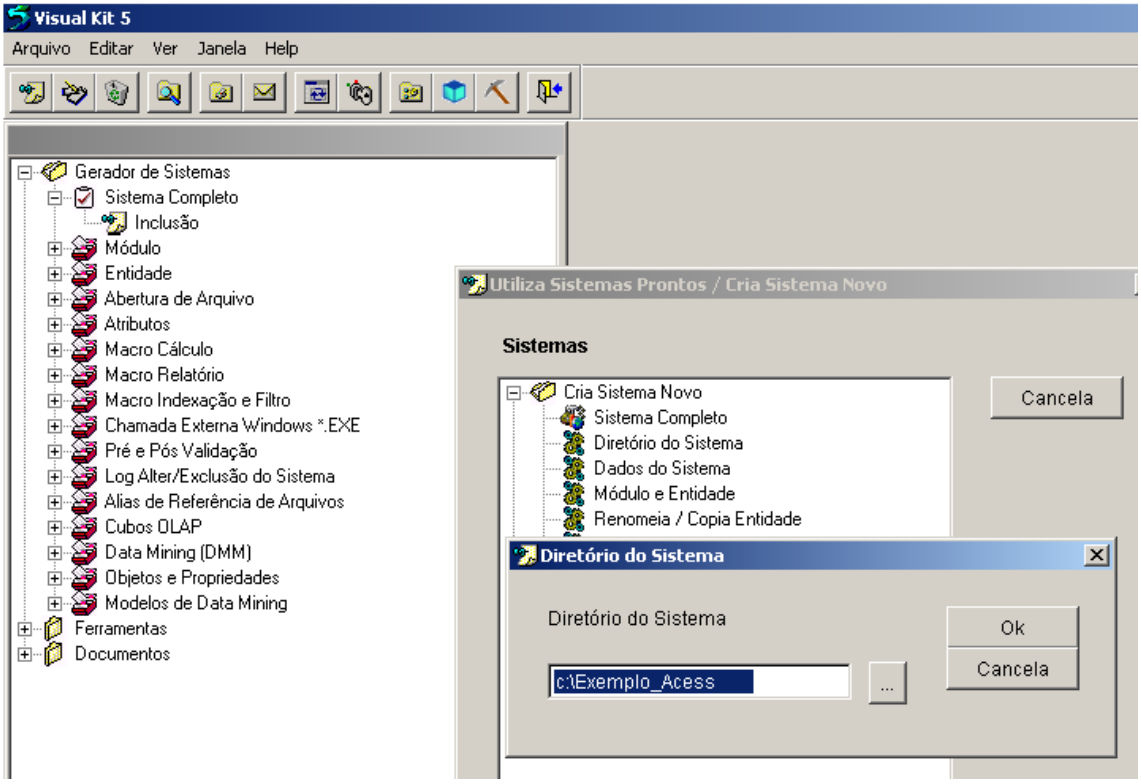


Fig 177

2) Crie um Módulo "**Cadastro**" e uma Entidade "**Clientes**" e depois feche o Visual Kit5 (ver Fig02)

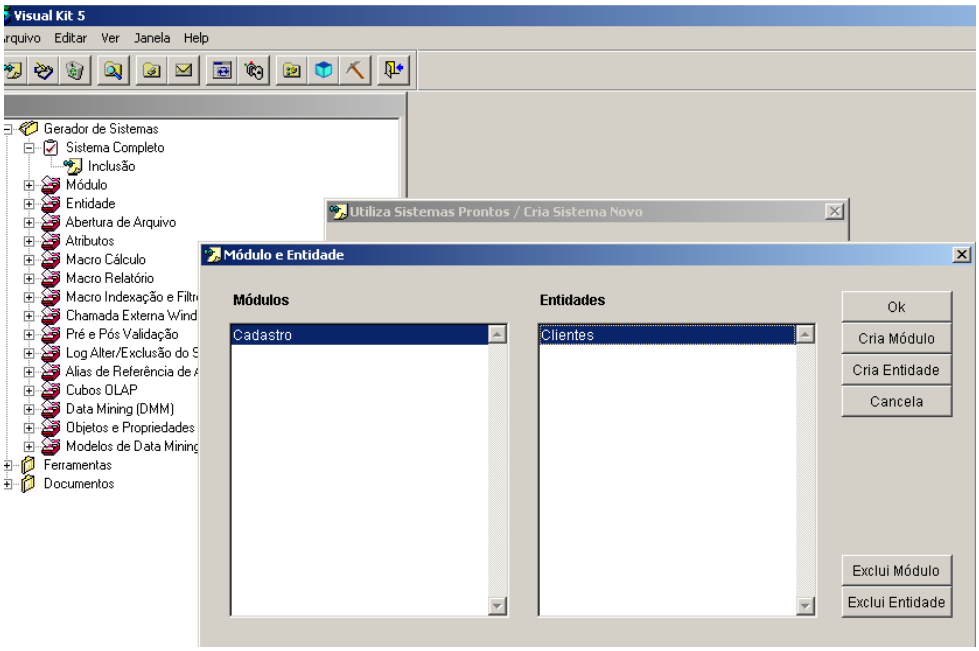


Fig 178

2) Entre na Pasta "**Exemplo_Acess**" ,clique na pasta "**DataBase**"
3) Exclua o arquivo "**mdbbase.mdb**" existente (ver Fig03)

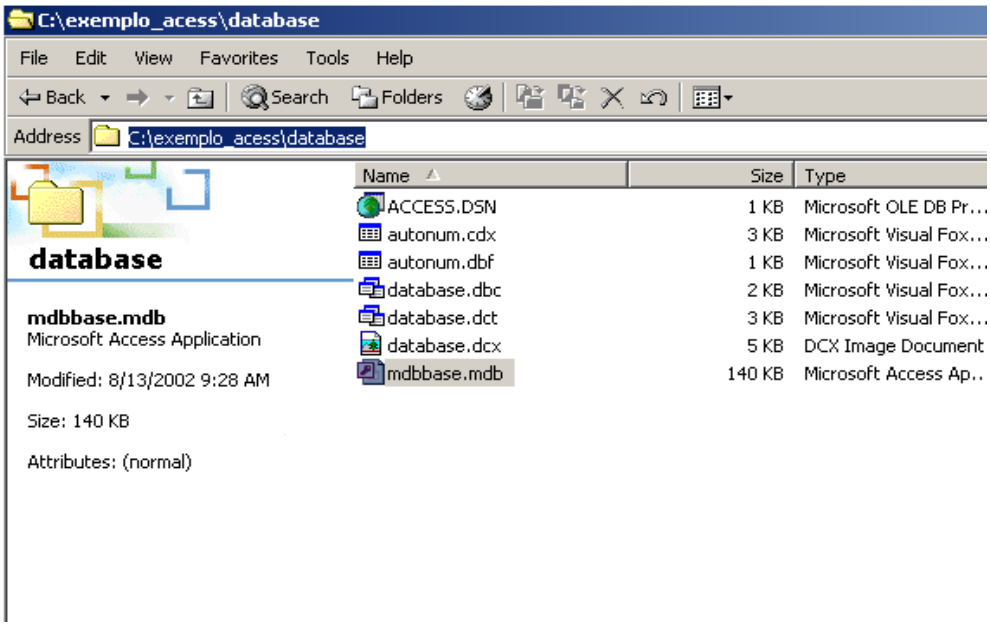


Fig 179

5) Salve o Banco criado por você em **Acess** com o nome de "**mdbbase.mdb**"

6) Entre no Visual Kit5 e depois clique em **"Abertura de Arquivo" >"Inclusão"**. Digite os seguintes dados: (ver Fig04)

Nº do Módulo = 01
Nº da Entidade = 001
Num_Ope = 1
Nova_Area = 001
Table_File = B01001
Index_Key = PK_CLIENTE
Index_File = B010011
Data_Base_ = C:\EXEMPLO_ACESS\DATABASE\MDBBASE.MDB
DB_DRIVE_ = 08
DB_SERVER_ = ACESS
PORT_SERV_ = 1433
WSID = LOCALHOST

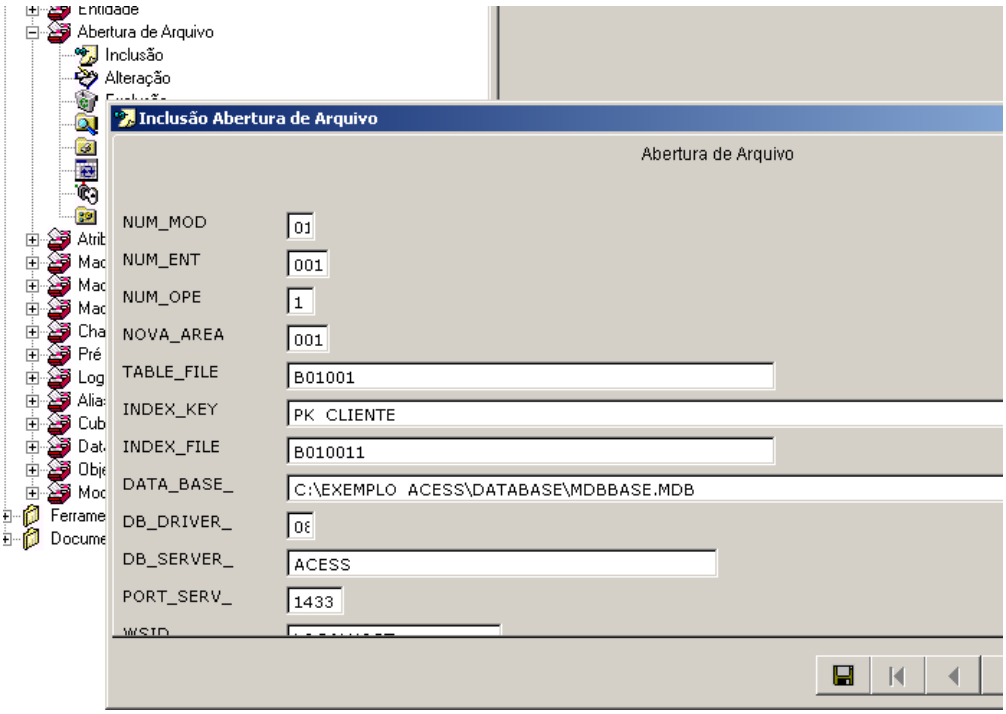


Fig 180

7) Salve e feche a Tela de "**Abertura de Arquivo**"

8) Clique em "**Alias de Referência de Arquivos">"Inclusão"**

9) Digite os seguintes dados:

- o número referente a Tabela.**TABLE_FILE**=B01001
- o nome da Tabela do Banco Acess criado que no exemplo é "**CLIENTES**"(ver Figura)

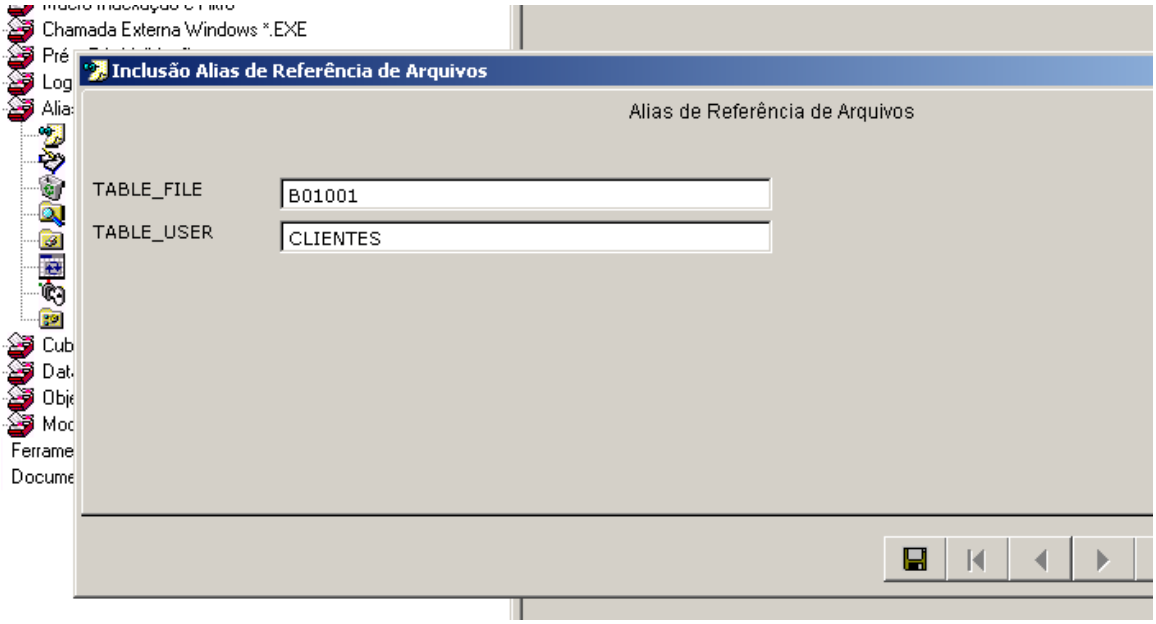


Fig 181

10) Salve e feche o Visual Kit5

11) Entre no Diretório **C** e localize a pasta criada (**c:\Exemplo_Acess**)

12) Clique em "**Principal.exe**"

Importando dados de uma Planilha Excel (*.xls)

1) Entre no Sistema Gerado por você, e escolha a **Entidade** que receberá os dados da planilha e clique na opção **Transferência**. (ver Fig01)

A entidade deve ter a mesma quantidade de campos que a tabela Excel. Lembre que geralmente na entidade o primeiro campo é chave. Por tanto coloque a primeira coluna da planilha Excel seja chave Exemplo um número sequencial.

2) Mude a opção **Formato Arq. Transferência** para **XLS – Excel** (ver Fig01)

3) Mude para Importa (para importar a planilha) (ver Fig01)
4) Clique em **Processar** (ver Fig01)
5) Digite o caminho onde está gravada a Planilha. Ex: **c:\Book1.xls**

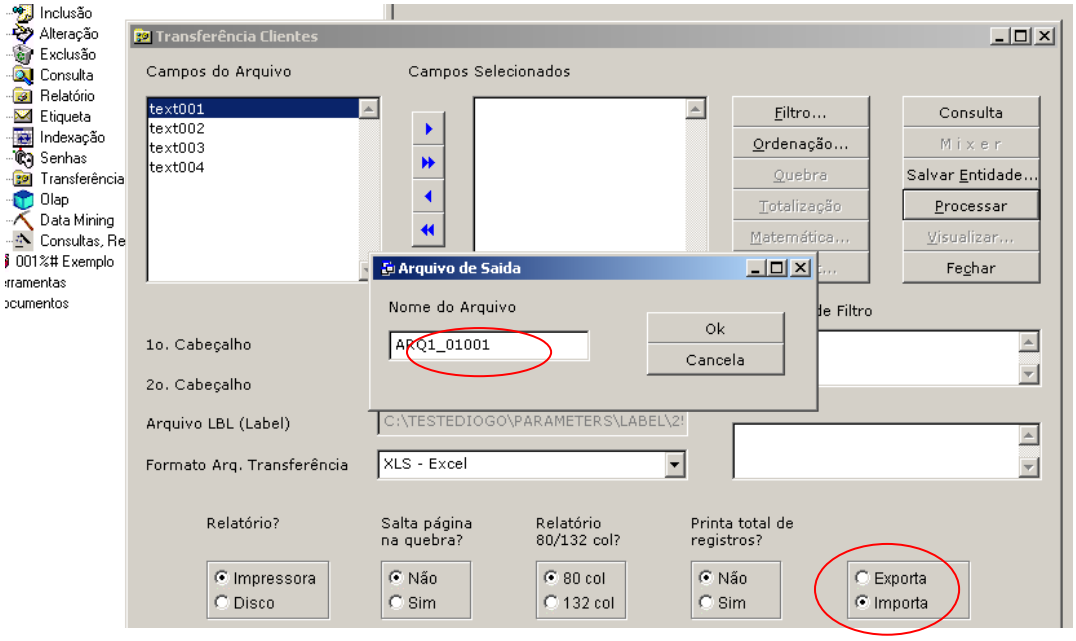


Fig 182

6) Clique em **Consulta** para conferir os dados na tabela
7) Clique em **Salvar Entidade** para que os dados sejam salvos na entidade desejada.
8) Digite o nome da **Entidade** e clique em **OK** (ver Fig02)

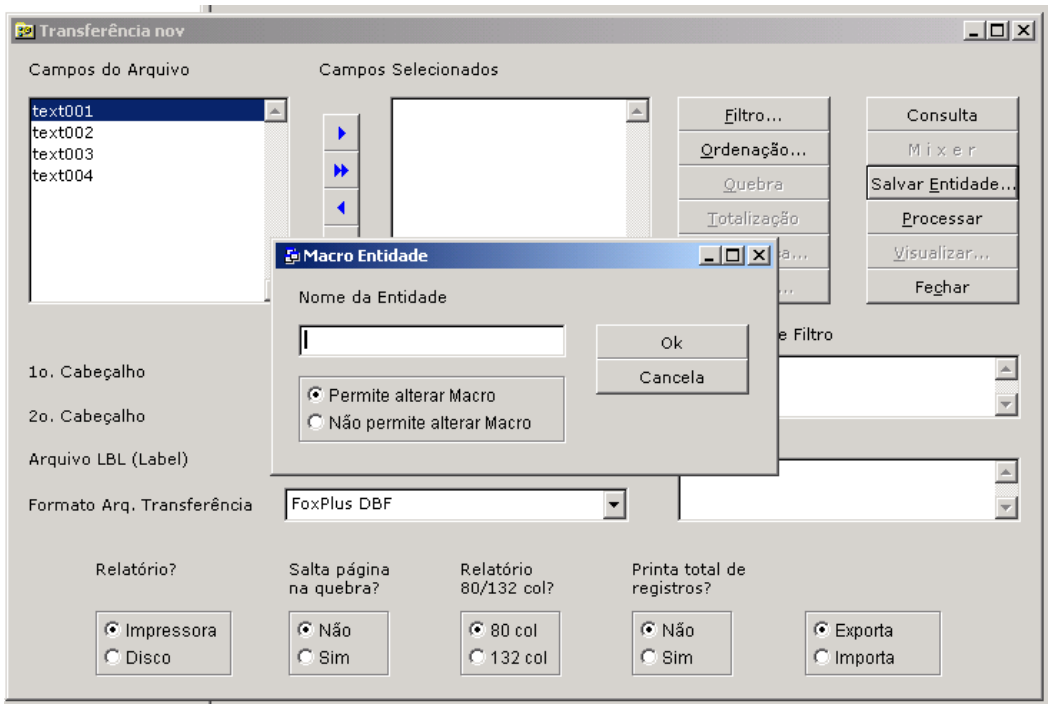

Transferência nov
Campos do Arquivo
Campos Selecionados
text001
text002
text003
text004
Filtro...
Ordenação...
Quebra
Totalização
Consulta
Mixer
Salvar Entidade...
Processar
Visualizar...
Fechar
Macro Entidade
Nome da Entidade
Ok
Cancela
Permite alterar Macro
Não permite alterar Macro
1o. Cabeçalho
2o. Cabeçalho
Arquivo LBL (Label)
Formato Arq. Transferência
FoxPlus DBF
Relatório?
Impressora
Disco
Salta página na quebra?
Não
Sim
Relatório 80/132 col?
80 col
132 col
Printa total de registros?
Não
Sim
Exporta
Importa

Fig 183

XIII. Instalando o Sistema Gerado em outro computador

Bom, ate aqui você aprendeu a desenvolver sistemas usando o Visual Kit 5 e testá-lo no seu próprio computador. Vamos agora, impressionar alguém, instalando um sistema gerado por você no computador de essa pessoa. Que tal?...

Instalando num único computador

Para executar um sistema gerado com o Visual Kit 5 na máquina de outro usuário você precisa realizar os seguintes procedimentos:

- Localize no CD-ROM do Visual Kit 5 o sub-diretório chamado **conf_usu** (veja Fig 184)**.** Nele você encontrará um programa executável chamado "**webapp.exe".** (Veja na Fig 185).

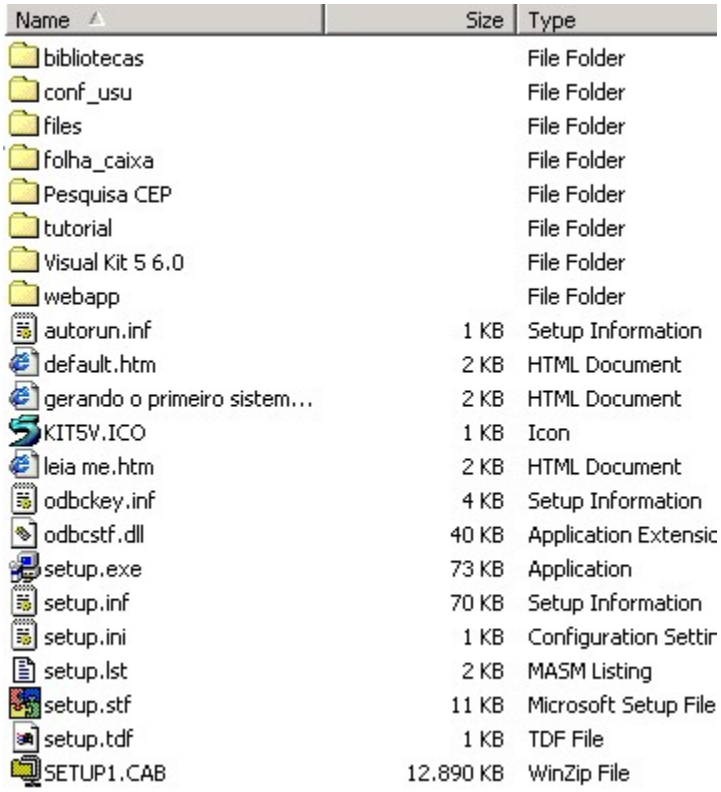


Fig 184 Conteúdo do CD Visual Kit 5

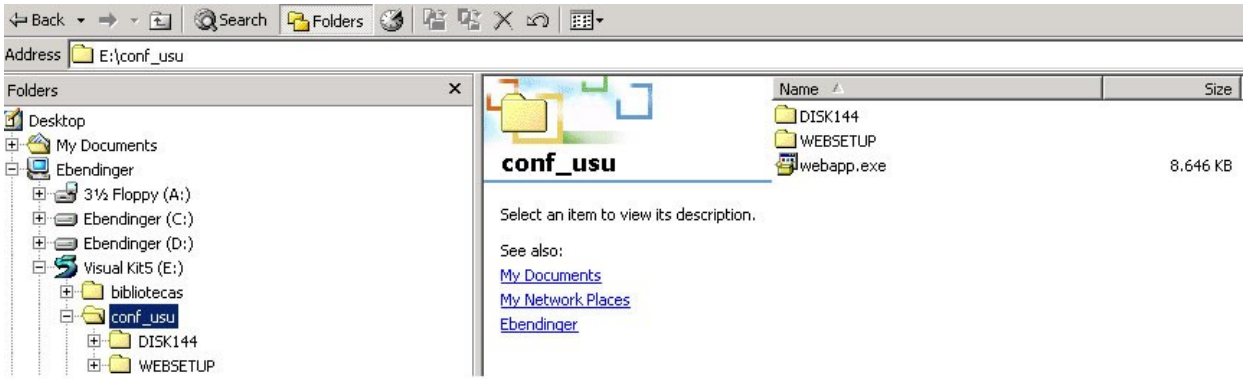


Fig 185Subdiretório CONF_USU do CD Visual Kit 5

- Execute o programa **webapp.exe** na máquina onde você vai instalar o sistema gerado, ou seja aquela pessoa que você quer impressionar. Este programa instalará todas as bibliotecas necessárias para o bom funcionamento do sistema gerado pelo Visual Kit 5;
- Uma vez instaladas as bibliotecas, você precisará simplesmente copiar o diretório completo do sistema gerado para o computador deste usuário final. E pronto!

Atenção!

Se você gravou o sistema gerado num CD-ROM para ser instalado no outro computador, não esqueça -depois de gravado no computador- colocar o atributo Ready-only (somente leitura) desativado. As instruções são as seguintes:

- Abra o programa *Explorer* do Windows. Clique com o botão direito do mouse encima da pasta do sistema gerado.
- Aparecerá uma janela como mostra a Fig 186

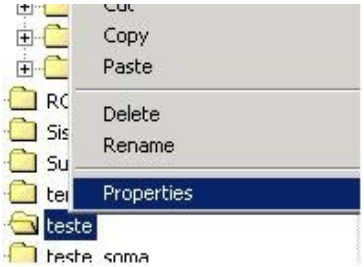


Fig 186 Janela no aberta no Explorer

- Selecione **Properties** (Propriedades).
- Será aberta uma nova janela com as diferentes propriedades do subdiretório.
- Localize o atributo Ready-only (somente leitura) e desative-o ficando como mostra a Fig 187.

Fig 187 Propriedades do subdiretório

Caso essa pessoa não tenha um leitor de CD, você pode copiar para 8 (oito) disquetes o instalador que se encontra no sub-diretório **\conf_usu\ disk144** do CD Visual Kit 5. E executar o instalador no computador do usuário final a partir de disquetes.

Nota:

Caso o sistema tenha funções específicas, você não precisa copiar os programas fontes (*.PRG) para o computador do usuário.

Instalando um Sistema Multiusuário

Os sistema gerados pelo Visual Kit 5 são sistemas multiusuários, ou seja você pode rodar o mesmo sistemas em vários computadores usando uma estrutura cliente-servidor.

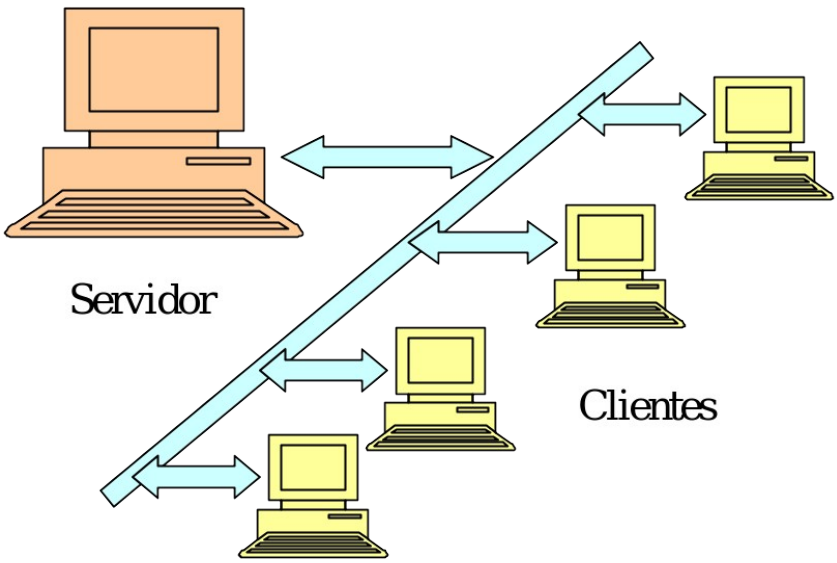


Fig 188 Esquema de estrutura Cliente-Servidor.

Segue os procedimentos para colocar um sistema gerado em uma rede de multiusuários.

- Copiar todo diretório do Sistema Gerado no computador servidor.
- Criar um mapeamento de rede nas estações de trabalho de cada usuário apontando para este diretório do servidor.
- Criar um diretório nas estações de trabalho.
- Copiar os seguintes arquivos: **principal.exe**, **vn5.app** e **config.fpw** que se encontram no diretório do sistema gerado, em cada diretório criado nas estações de trabalho do passo anterior.
- Abrir o arquivo **CONFIG.FPW** de cada estação de trabalho usando o programa **Notepad** (Bloco de Notas) -veja na Fig 189- para fazer uma simples modificação.

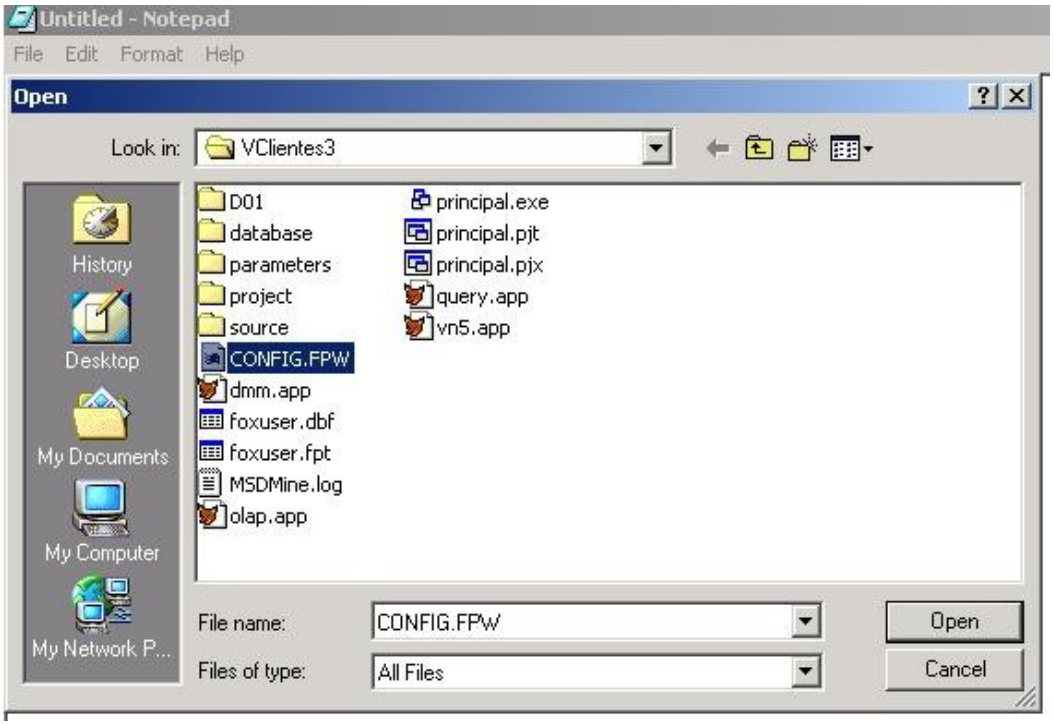


Fig 189 Abrindo o arquivo CONFIG.FPW pelo Notepad.

O arquivo **CONFIG.FPW** dos sistema gerados tem o formato mostrado na Fig 190 . Mude a letra **C:** que indica o *path* para a letra em que foi feito o mapeamento de rede nas estações de trabalho.

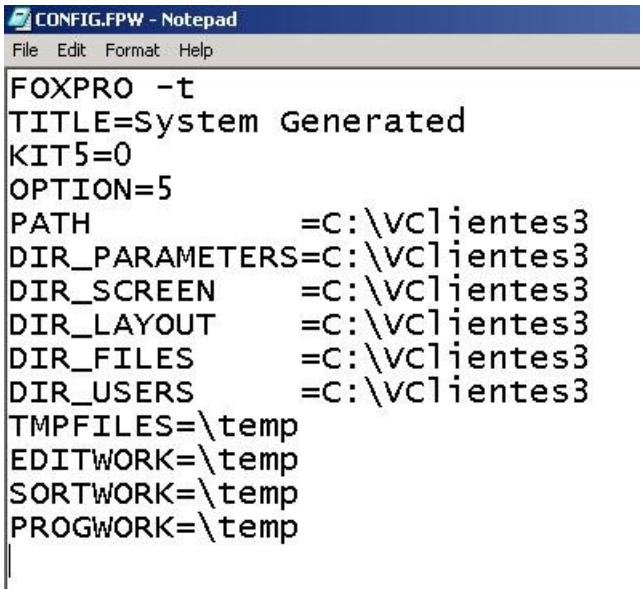


Fig 190 Arquivo CONFIG.FPW inicial

Por exemplo, se o mapeamento foi feito usando a letra **H:\,** o *path* no arquivo CONFIG.FPW ficará da seguinte forma:

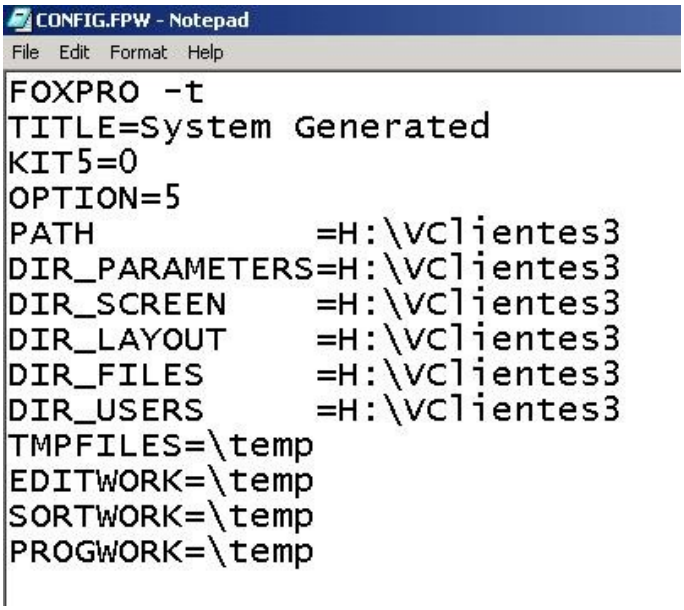


Fig 191 Arquivo CONFIG.FPW modificado.

XIV. Voltando ao Ambiente de Desenvolvimento

No Capítulo III descrevemos brevemente o ambiente de desenvolvimento do Visual Kit 5. Agora que já você tem certa experiência com o a ferramenta, voltemos descrever determinadas opções muito úteis quando desenvolvemos sistemas.

Renomeia e Copia Entidades

É comum, que durante o desenvolvimento de um sistema, você precise mudar uma entidade de módulo, ou simplesmente renomea-la ou aproveitar uma entidade já desenvolvida em outro sistema e copiá-la para o sistema que está em desenvolvendo, etc. Todas estas opções estão disponíveis no ambiente de desenvolvimento do Visual Kit 5, vejamos como proceder:

- Clique na *TreeView* para abrir a pasta “Gerador de Sistema”, logo em “Sistema Completo” e selecione a opção “Inclusão”.
- Aparecerá a janela Utiliza Sistemas Prontos / Cria sistema Novo (veja na Fig 192). Selecione a opção **Renomeia / Copia Entidade**.
- Será aberta uma janela intitulada **Renomeia / Copia Entidade** (veja na Fig 193)

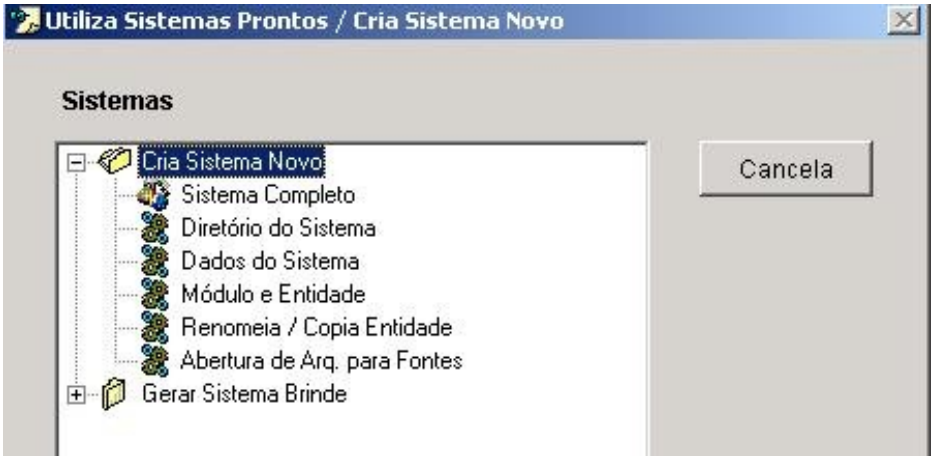


Fig 192 Acessando a opção Renomeia / Copia Entidade

Para renomear ou copiar entidades basta preencher as entradas de dados desta janela. Tendo sempre em conta que na entrada Módulo e Entidade você colocará a numeração o padrão usada pelo Visual Kit 5. (veja no apêndice C)

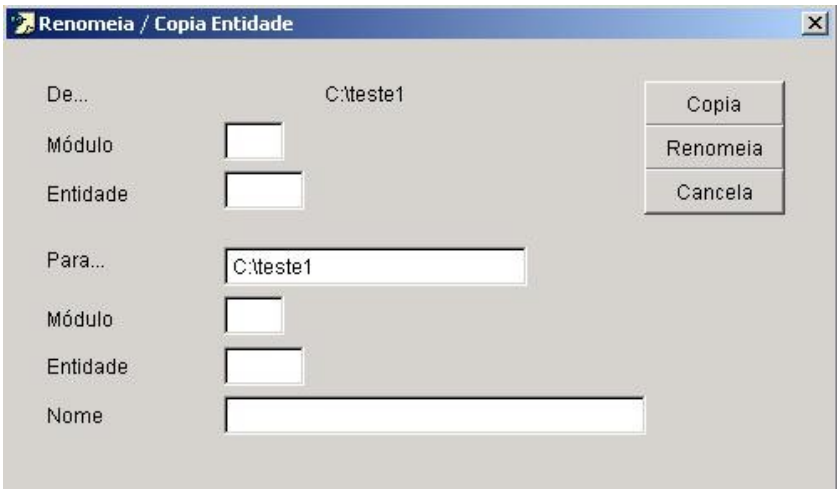


Fig 193 Janela Renomeia e Copia Entidade

Nota

Você poderá renomear e/ou mudar as entidades dentro de um mesmo sistema através da opção "Entidade->Consulta" no menu *TreeView* do ambiente de desenvolvimento. Esta opção será descrita ainda neste capítulo.

Vejamos o exemplo da Fig 194, aqui estamos copiando a Entidade: 001 do Módulo: 01 do sistema "*teste1*" para o módulo 02 entidade 003 do mesmo sistema.

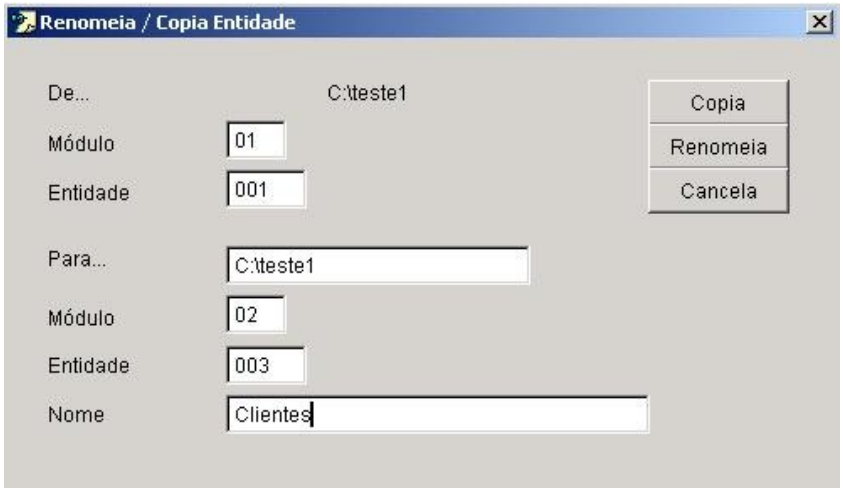


Fig 194 Copiando entidades dentro do mesmo sistema

Atenção!

Está opção não permitirá copiar encima de uma outra entidade já existente, por isso verifique primeiro se no módulo receptor não existe uma entidade com o mesmo número.

Para copiar entidades para outros sistemas - e desta forma aproveitar entidades prontas já desenvolvidas - basta colocar o nome do sistema na entrada de dados "**Para...".** Por exemplo "teste2" – veja na próxima figura. Aqui será copiada a entidade 001 do módulo 01 do sistema "teste1" para o módulo 01, entidade 002 do sistema "teste2".

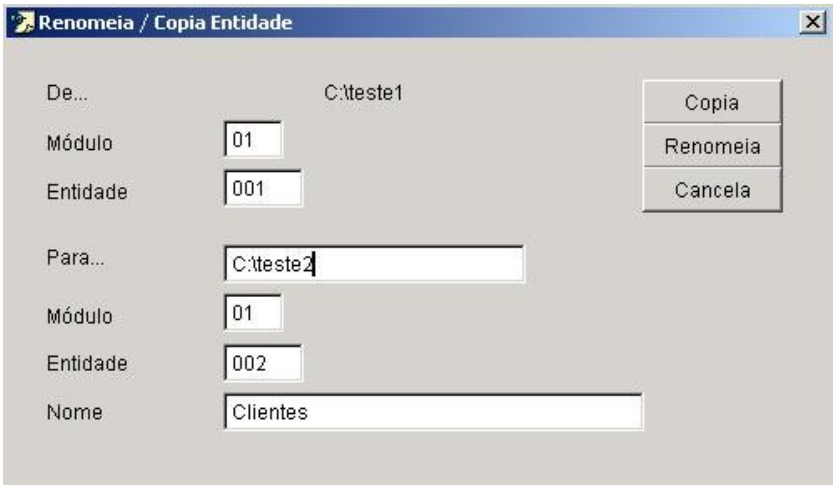


Fig 195 Aproveitando entidades prontas.

Atenção!

Quando copiamos entidades usando esta opção, estaremos copiando somente a estrutura da entidade, ou seja, o desenho do formulário, as propriedades e validações e nunca os dados cadastrados.

Se você precisa copiar também os dados cadastrados, precisará realizar uma exportação/importação de dados.

XV. Criando Funções pelo FoxPro

Como temos visto ate agora, o Visual Kit 5, têm uma ampla gama de funções e recursos para desenvolver sistemas. No entanto, você pode desenvolver novas funcionalidades utilizando a linguagem de programação *Microsoft Visual FoxPro for Windows versão 6.0*.

Neste capitulo abordaremos os procedimentos a ser feitos para desenvolver funções no *FoxPro* e embuti-las no Visual Kit 5, portanto vai requer conhecimentos de programação. Se você não é programador, pule por favor este capítulo.

Você poderá criar tanto funções especificas associadas as operações Inclusão, Exclusão, Alteração, etc. como funções de Pré ou Pós-Validação de campos.

Criando funções específicas.

Existem dois tipos de funções específicas que podem ser criadas:

- *Função Com Abertura de Arquivos*: Esta função abre os arquivos que você tenha especificado previamente no Visual Kit 5, no menu “Abertura de Arquivos” e em seguida executa o programa que você desenvolveu em Visual FoxPro.
- *Função Sem Abertura de Arquivos*: Não abre nenhum arquivo definido pelo Visual Kit 5 - é totalmente independente.

A continuação descrevemos os procedimentos para criar as função especificas.

- Execute o Visual Kit 5.
- Crie a entidade que terá como base a função específica. Veja no exemplo da Fig 198.

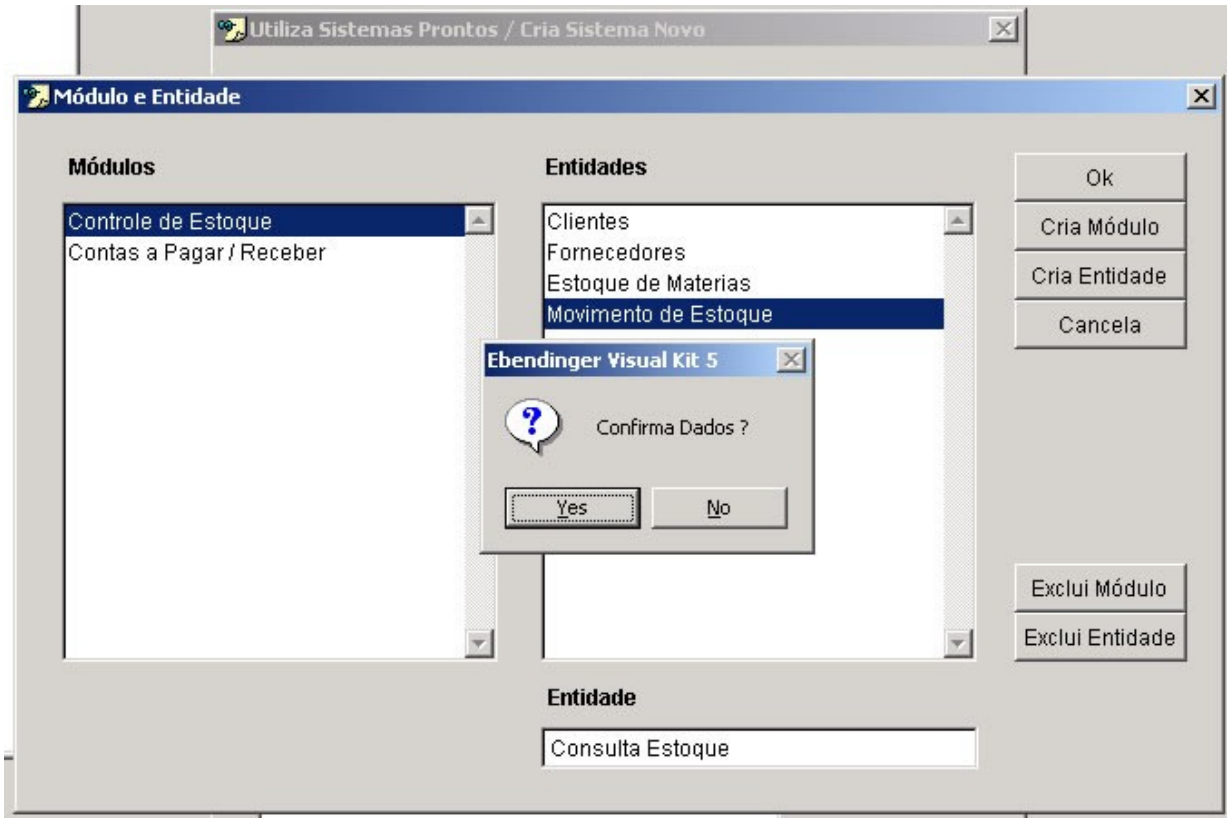


Fig 196 Criando Entidade para função especifica.

- Na janela de **Entidades**, selecione para a operação **Inclusão** o tipo de função especifica: com abertura ou sem abertura de arquivos. Para as restantes operações desabilite as funções básicas do Visual Kit 5. Veja a Fig 197.
- Pressione o botão **Ok**, para criar a função específica.
- O Visual Kit 5 criará automaticamente um arquivo com nome que segue uma regra especifica que a continuação descreveremos. Este arquivo receberá o código da função que você deseja incluir
- Indicará também em qual diretório se encontra este arquivo, como mostra a Fig 198.
- Pressione o botão **Ok**, para confirmar.

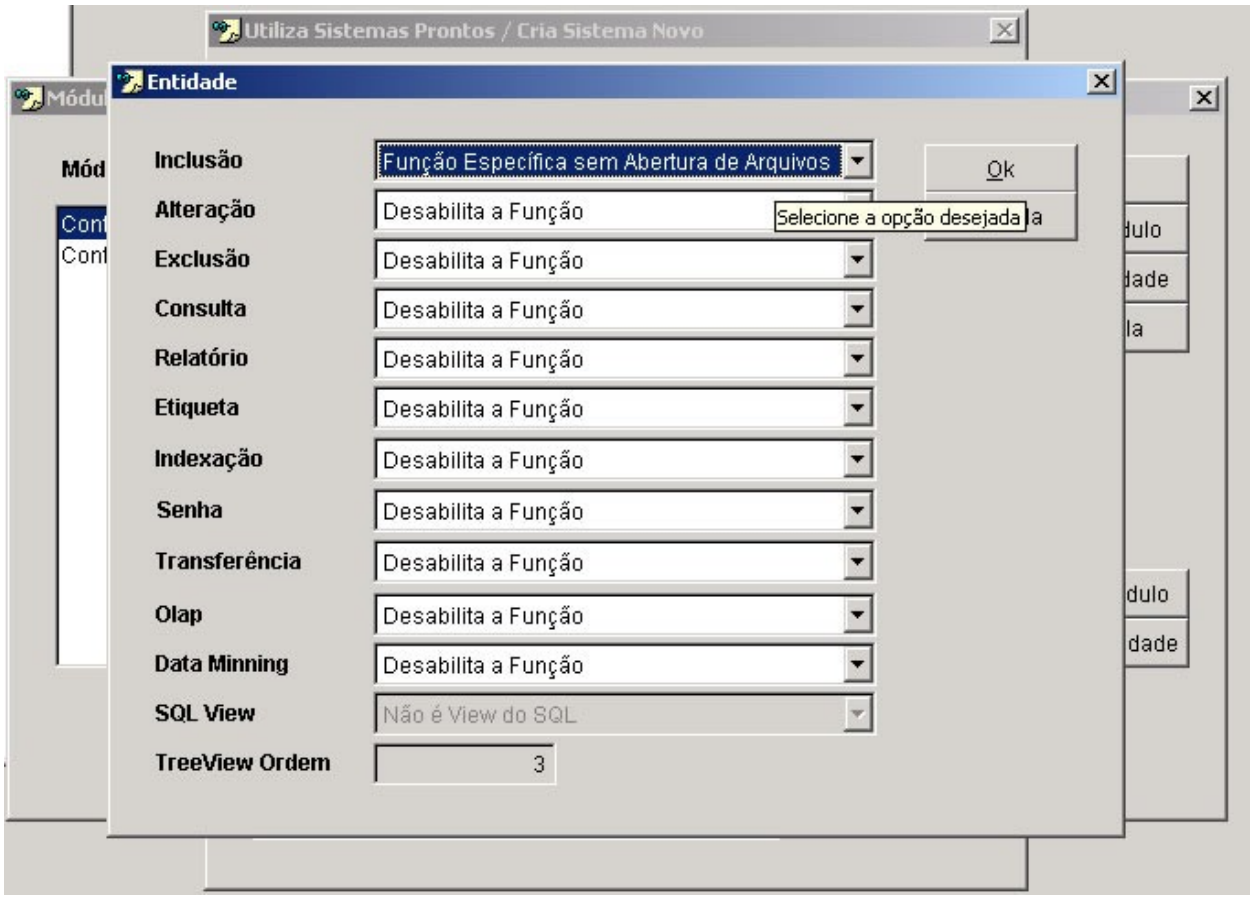


Fig 197 Escolhendo o tipo de função especifica.

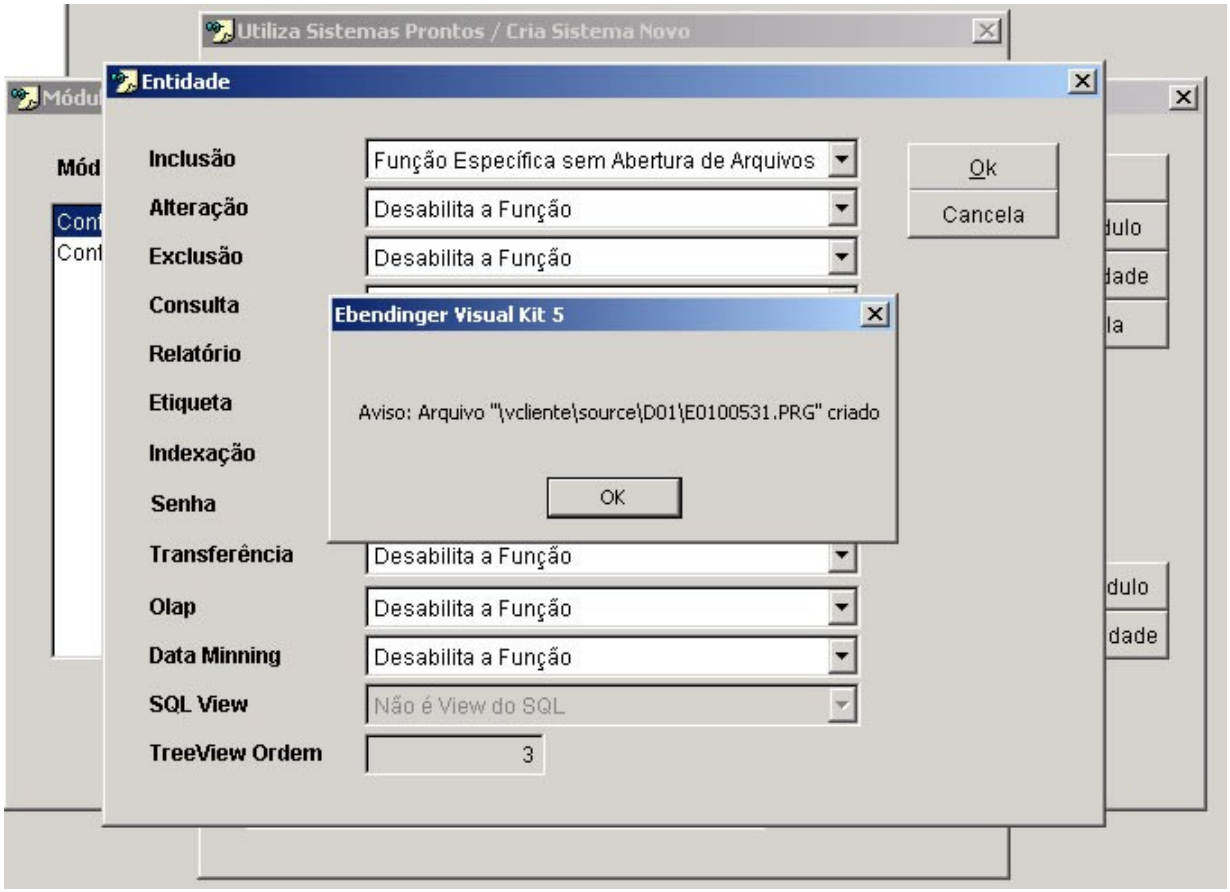


Fig 198 Indicando o arquivo da função especifica.

- Utilizando o Visual FoxPro localize o projeto do sistema gerado e dentro dele o arquivo gerado (.PRG);
- O Visual Kit já inclui códigos iniciais neste arquivo gerado. Veja Fig 200;
- Complemente com códigos de programação o conteúdo de este arquivo.
- Insira no projeto PRINCIPAL.PJX o programa (.PRG) que você criou;
- Compile o programa;
- Agora, você pode executar o sistema gerado com a função específica embutida em seu código executável.

Nota:

Sempre que é criada uma entidade do tipo “função específica”, o Visual Kit 5 cria no diretório \<sistema gerado>\source\<número do módulo> o arquivo (.PRG) vazio contendo o nome codificado.

No Visual FoxPro v.6.0 para:

- Abrir o projeto PRINCIPAL.PJX utilize o menu “Arquivo”, opção “Abrir”;
- Adicionar o programa (PRG) utilize o botão “Adicionar”
- Modificar o programa (PRG) utiliza o botão “Modificar”;
- Compilar o projeto usa o botão “Construir”

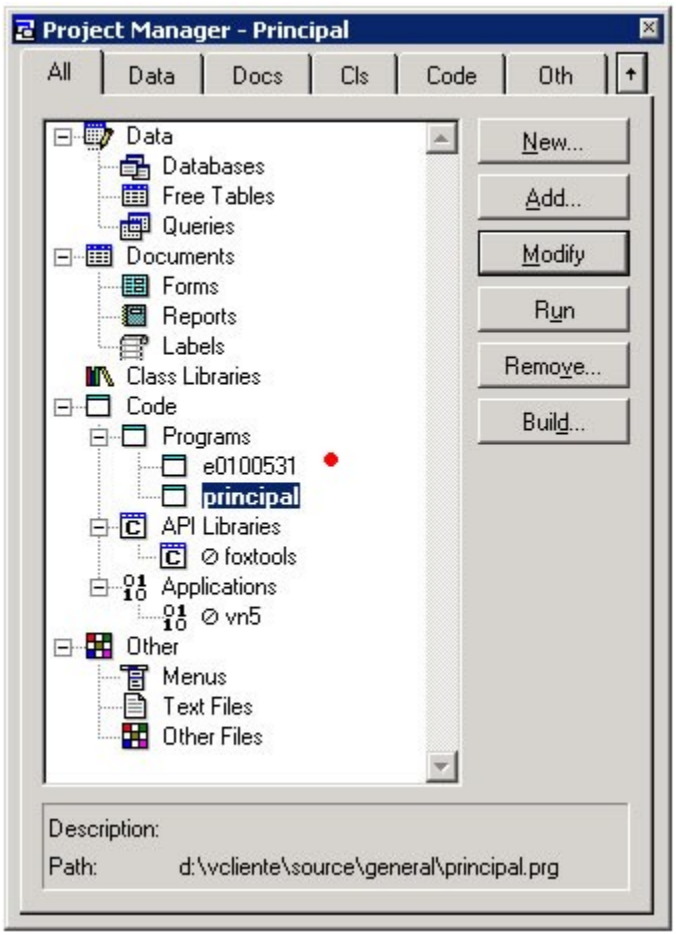


Fig 199 Arquivo gerado automaticamente pelo Visual Kit 5 para inclusão do código da função especifica.

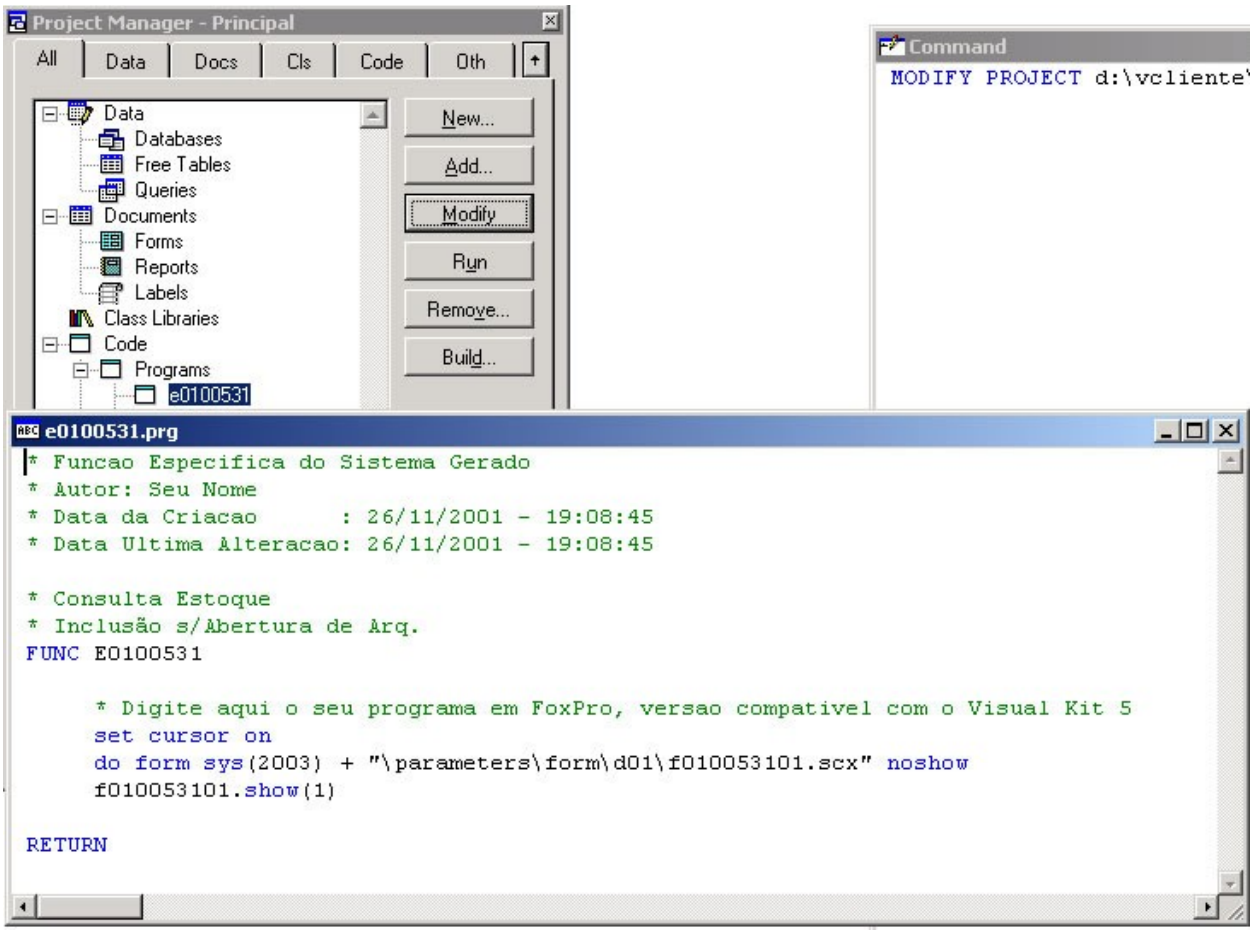


Fig 200 Visão do Arquivo criado listo para a inclusão de códigos.

Regra do nome da função específica

O Visual Kit 5 denomina as funções especificas seguindo um determinado padrão. Veja no seguinte exemplo

FUNCTION E0100121()
comandos...
comandos...
RETURN

- O primeiro caractere do nome da função sempre é letra "**E**".
- Os próximos dois dígitos indicam o número do módulo, Ex 01, 02, 03;
- Os próximos três dígitos indicam o número da entidade. Ex: 001, 002, 003

- O próximo dígito indica o tipo de função específica: "2" para função específica com abertura de arquivo; "3" para função específica sem abertura de arquivo; "1" para operação básica mais função específica.
- O ultimo digito indica o número da operação básica que irá chamar a função específica:
  "1" para "Inclusão";
  "2" para "Alteração";
  "3" para "Exclusão";
  "4" para "Consultas";
  "5" para "Relatórios";
  "6" para "Etiquetas";
  "7" para "Indexação";
  "9" para "Transferência".

Cuidados na manipulação de arquivos

O Visual Kit 5 gera sistemas multiusuário, por isso, tome alguns cuidados na manipulação de arquivos para evitar problemas com o sistema gerado. Quando uma função específica é executada o controle total do programa será efetuado pela sua função, e não pelo sistema gerado.

Para desenvolver funções específicas com manipulação de dados em arquivos, observe com atenção os seguintes comandos:

set exclusive (on/off)
Determina se a base de dados e os arquivos de índices serão abertos de forma compartilhada (off) ou exclusiva (on). O valor default deste comando nos sistemas gerados pelo Kit 5 é off (arquivos compartilhados).

set printer to <destino>
Determina o destino (alvo) da impressão. A impressão pode ser enviada para a impressora ou arquivo.

unlock [all]
Libera o travamento lógico de arquivos e registros na *Work Area* que foram marcados pelas funções de travamento.

flock()
Posiciona o arquivo para uso exclusivo do terminal que solicitou o travamento lógico. Retorna .t. se obteve sucesso.

rlock()
Posiciona o registro para uso exclusivo do terminal, que solicitou o travamento lógico. Retorna .t. se obteve sucesso.

use <arquivo> index <arq. indice> exclusive
Este comando deve ser utilizado para abrir um arquivo de dados e associá-lo a um arquivo de índice para uso exclusivo. Qualquer outro usuário que tente o travamento lógico deste arquivo ou de um registro do mesmo não obterá sucesso.

Para os comandos: **pack, reindex, zap, index on** os arquivos deverão ser abertos sempre em modo exclusivo:

Nota:

- Sempre que uma operação de *escrita* for necessária (*replace*), deve ser efetuado o travamento lógico. Para as outras situações é opcional;
- NUNCA coloque o arquivo de índice na sintaxe do comando USE;
- O campos GERAL (imagem) tem que ser o último campo do arquivo.

Criando funções de validação de campos

Como vimos anteriormente, as funções de validação de dados são as funções chamadas pelas Pré e Pós-Validação dos campos. O Visual Kit 5 possui varias funções de validação, mas você pode ampliar este número criando outras - usando o Visual FoxPro.

Você deverá abrir o projeto do Sistema Gerado pelo FoxPro e criar o código da função. Está função acessará os valores dos campos através do vetor *vdados.* Veja no seguinte exemplo a forma de como fazer isto:

```
FUNCTION mult5
        external array vdados
        vdados[I] = vdados[I]*5
RETURN .t.
```

Compile o projeto pelo Visual Kit 5 e incluía na Pré ou Pós-Validação a função criada, exemplo: *mult5()*

XVI. Criando um Sistema no MS SQL Server 2000.

Neste capitulo abordaremos como criar um sistema no Banco de Dados MS SQL Server 2000 usando o Visual Kit 5.

O SQL Server

Sai do marco deste manual a explicação do que é o MS SQL Server e como usá-lo, por isso faremos só uma breve descrição – maiores informações você poderá encontrar em muitos livros especializados.

O SQL Server é um sistema de gerenciamento de bancos de dados cliente/servidor de alto desempenho com alta integração com o Windows NT. Suas características são: Integração com os serviços de *multithreading* [múltiplas linhas]; agendamento; Monitor de Desempenho; log de eventos do Windows NT.

Com uma replicação nativa permite disseminar informações para vários locais, reduzindo a dependência de um servidor único, e deixando a informação necessária mais próxima de quem realmente precisa dela. Tem uma arquitetura paralela, que executa as funções de banco de dados simultaneamente para diversos usuários e tira proveito de sistemas com múltiplos processadores.

Possui um gerenciamento centralizado de todos os servidores através de uma arquitetura de gerenciamento distribuída, com uma interface visual.

SQL Server- Visual Kit 5

Antes de criar um sistema pelo Visual Kit 5 no SQL-Server, é necessário cumprir determinados requisitos:

- Ter já uma Base de Dados (*Database*) criada no SQL-Server
- Ser usuário cadastrado no *Database.*

- Ter permissão de criar tabelas no *Database*.

Nota:
O Visual Kit 5 não cria diretamente o *Database*, pois envolve questões de segurança cujo critério varia de administrador para administrador do Base de Dados.

Após certificar – que você é usuário cadastrado e tem permissão de criar tabelas no *Database*, deve – se criar ou cadastrar o mesmo usuário (*login*) e senha no ambiente do Visual Kit 5, para isto basta:

- Ir em Ferramentas no menu *TreeView* do Visual Kit 5
- Selecionar Propriedade;

Aparecerá uma janela (veja na Fig 201) confirmando a criação do *login* e senha do administrador do Visual Kit 5. Clique em YES.

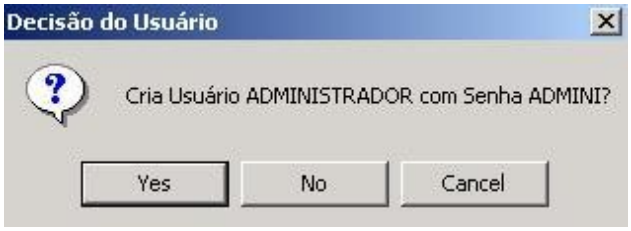


Fig 201 Criando usuário ADMINISTRADOR

Crie -como administrador do Visual Kit 5- um novo usuário com login e senha igual ao cadastrado no SQL-Server.

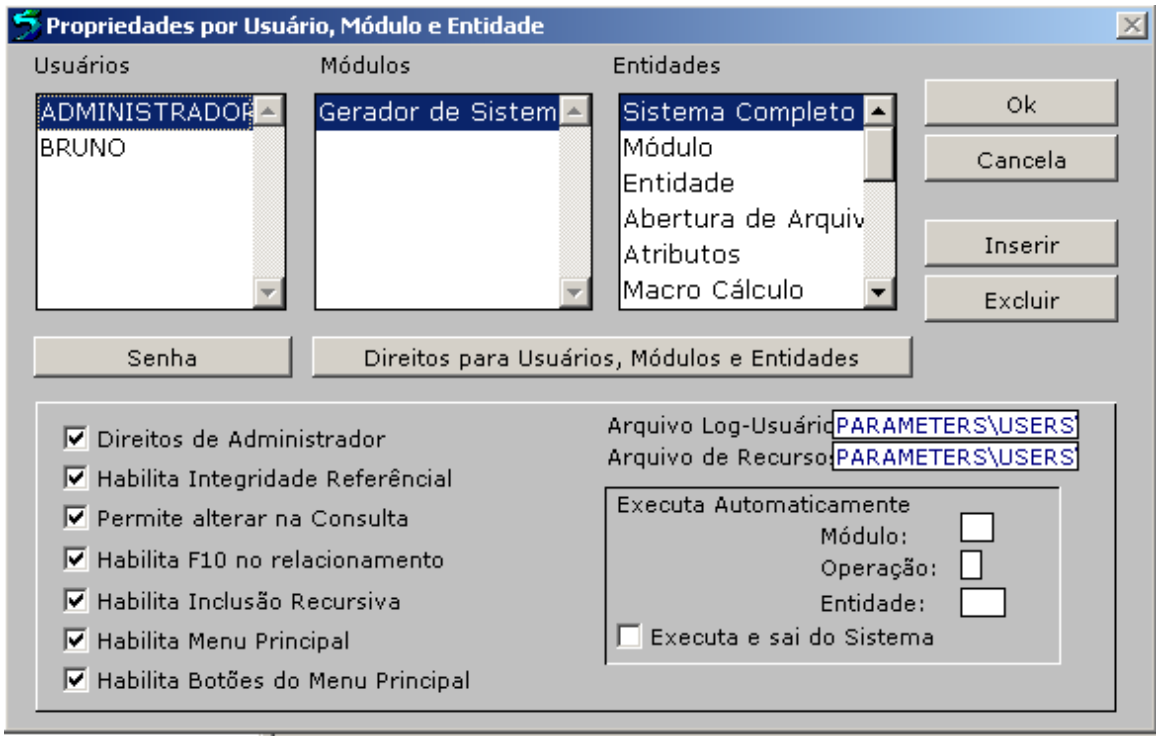


Fig 202 Incluindo novo usuário e senha no Visual Kit 5

Após isto feito, tudo está pronto para começar a desenvolver sistemas no MS SQL-Server. Como você já aprendeu:

- Clique em Sistema Completo -> Inclusão -> Cria Sistema Novo -> Sistema Completo e crie o novo diretório do sistema.
- Na janela de Dados do Sistemas, preencha os dados gerais do sistema, com uma única diferença, selecione na opção **Driver** -> Microsoft SQL-Server ao invés do Visual FoxPro.

Imediatamente, a janela será expandida para que se possa inserir configurações adicionais para a conexão com o SQL-Server.

Fig 203 Configurando a conexão com o SQL-Server.

Na entrada **Base de Dados** deverá ser configurada com o nome do *Database* dado pelo Administrador do SQL-Server que indica em qual instância o Visual Kit 5 irá criar as tabelas.

A entrada **Servidor** será configurada com o nome ou **IP** da máquina onde está localizado o Banco de Dados, caso o banco de dados seja local, deve – se botar o nome da máquina a qual se está utilizando. Caso o banco de dados esteja remoto via internet, deve – se obrigatoriamente colocar o IP da máquina ao invés do nome dela.

No **Porta do SQl** também deve ser solicitada ao Administrador, porém já tem o valor *default* com a porta padrão que é a **1433**.

A **WSID** indica o nome da Estação de Trabalho (Workstation) e não é necessário solicitar ao administrador, pode – se inserir qualquer informação, até mesmo o seu nome.

Após estes passos, a utilização do Visual Kit 5 volta a ser como de usual. A única diferença é que ao invés de criar arquivos em formatos .DBF ele criará tabelas no SQL-Server.

A modo de exemplo vamos criar, um sistema de um módulo e uma entidade. Lembremos que, o Visual Kit 5 enumeras as tabelas das

entidades usando cinco dígitos -seguindo a ordem de criação- cumprindo sempre o seguinte padrão: Começa sempre com a letra “B”, seguido de dois dígitos que indicam o numero do módulo e três dígitos que indicam o numero da entidade. Ex. B01001 – é a primeira entidade do primeiro módulo.

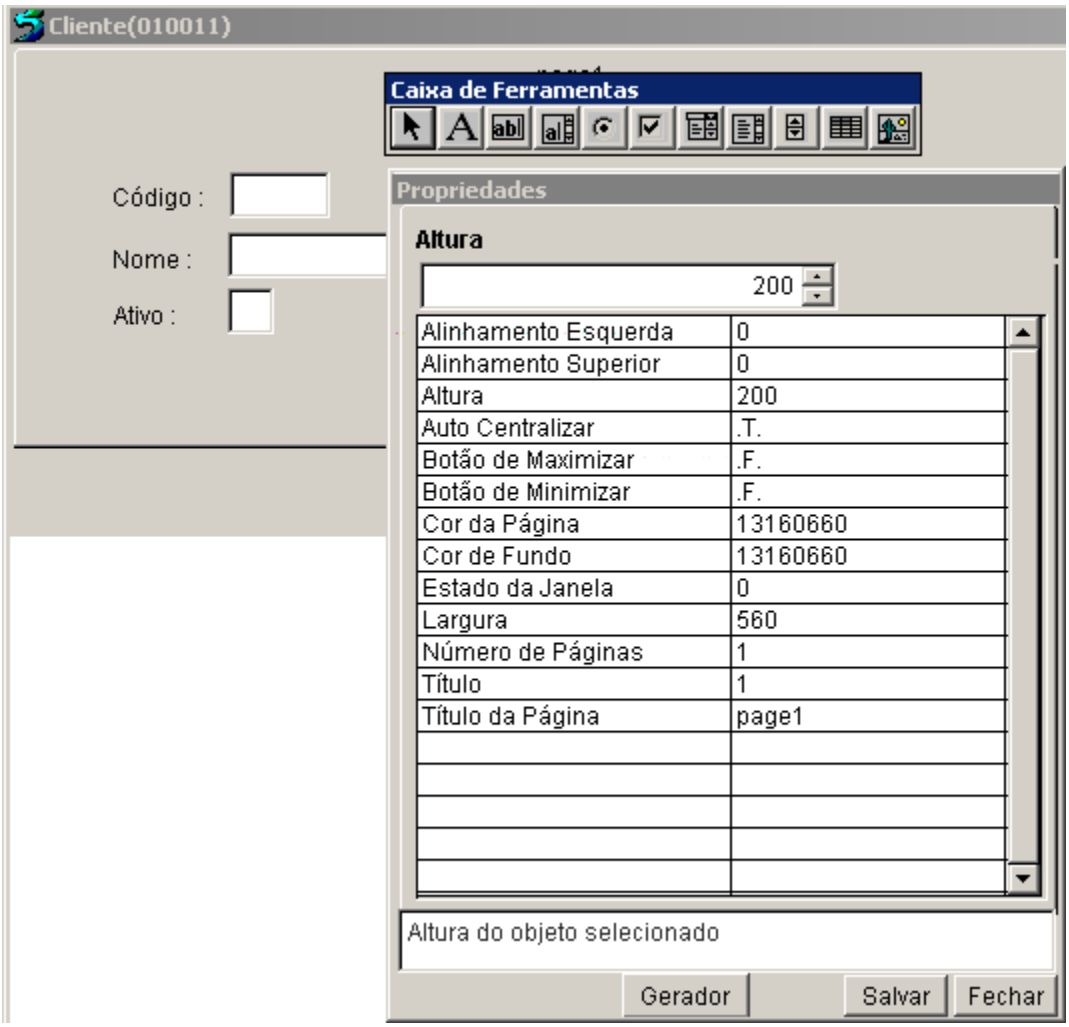


Fig 204 Criando a entidade B01001

Salvemos nosso sistema e entremos no SQL-Server. Deverá estar criada uma nova tabela: a B01001. (veja na Fig 205)

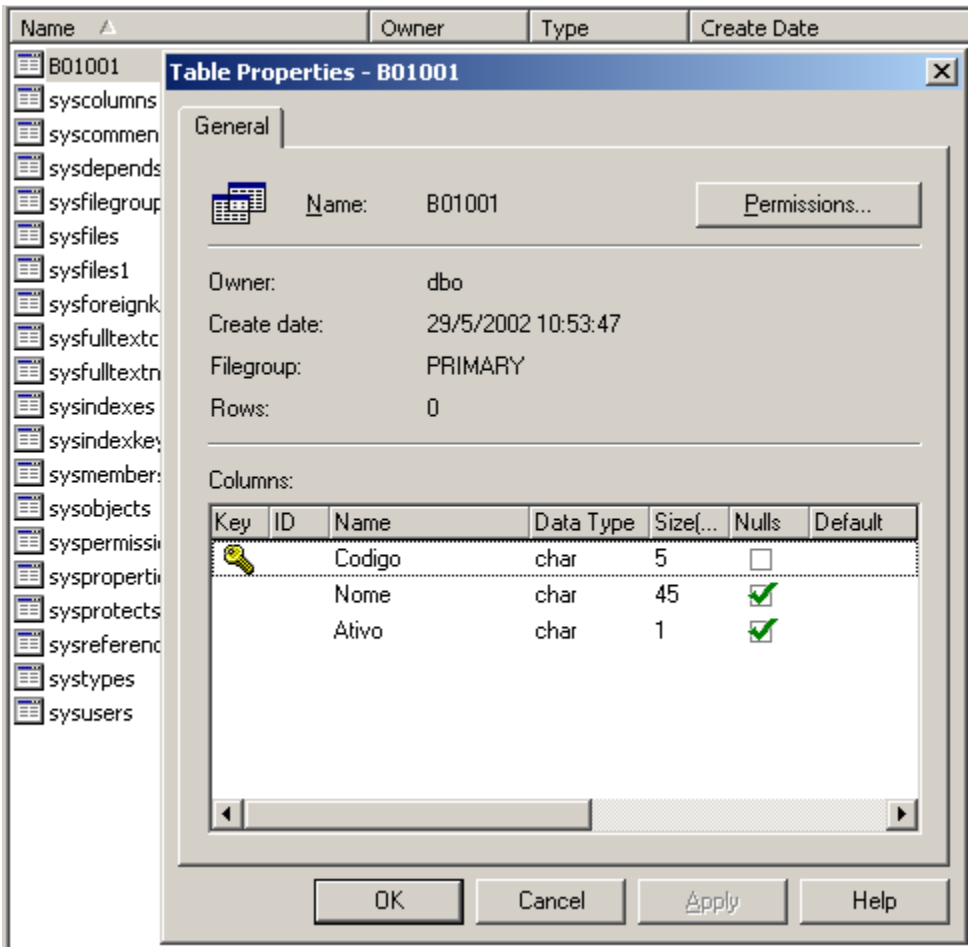


Fig 205 Tabela criada pelo Visual Kit 5 no MS SQL – Server.

Criando sistemas com conexão remota via internet.

Esta seção é só para ressaltar o dito anteriormente, quando a usamos Bases de Dados remotas pela internet devemos na janela Dados do Sistema, na indicação **Servidor** colocar um IP válido (exemplo: 200.224.132.102) aberto para acesso na porta 1433 e suas bases estarão acessíveis via internet, no entanto o seu sistema estará rodando no sua estação de trabalho.

Rodando pelo navegador de internet

Você pode rodar o sistema gerado pelo Visual Kit 5 usando o *Internet Explorer Browser*. Para isto você precisa fazer uma compilação usando o Visual FoxPro. Veja os procedimentos:

- Abra o projeto **principal.pjx** do sistema gerado pelo Visual FoxPro. Veja na Fig 206.

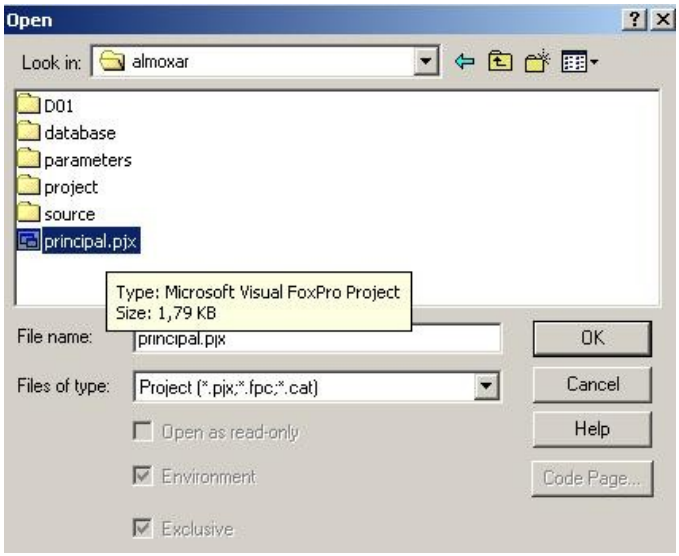


Fig 206 Abrindo o projeto pelo Visual FoxPro.

- Na janela **Project Manager** aperte o botão **Build**
- Será aberta a janela **Build Options**. Selecione em **Build Action** a opção **Application(.app)**. Confirme a seleção clicando o botão **OK**.

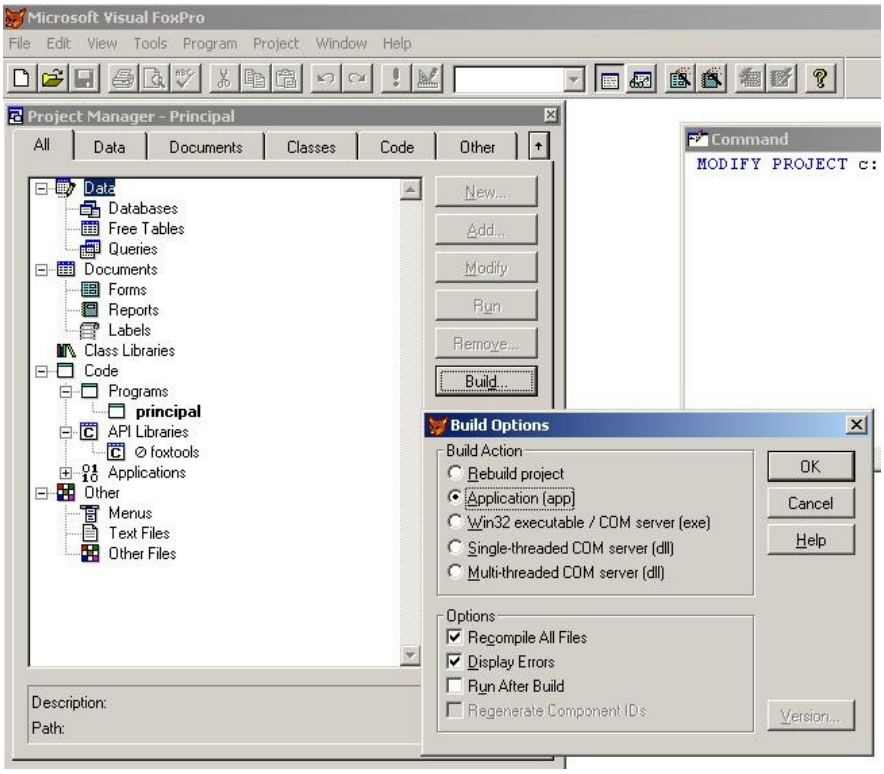


Fig 207 Selecionando opção Application (app).

Seguidamente aparecerá a janela de **Save as**, para confirmar o nome do arquivo **.app**. (veja na Fig 208).

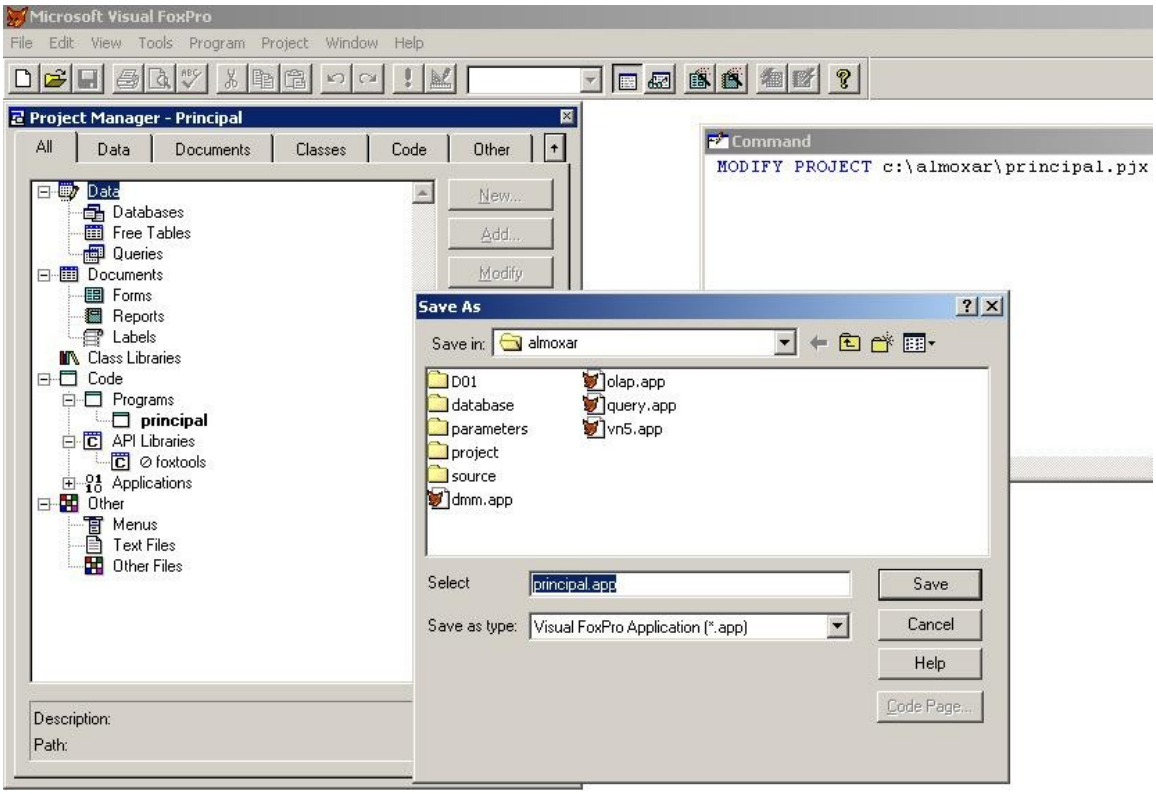


Fig 208 Confirmando nome do arquivo principal.app

O Visual FoxPro criará o arquivo **principal.app** dentro do diretório do sistema gerado. Veja no exemplo da Fig 209.

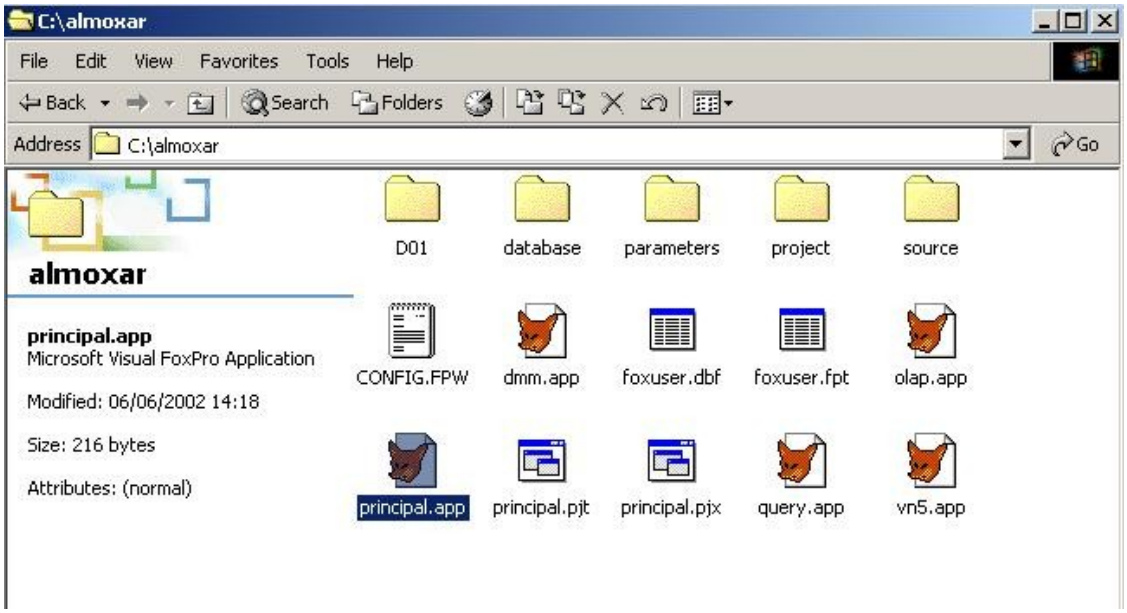


Fig 209 Arquivo principal.app criado

Agora é só abrir o **Internet Explorer** e localizar este arquivo no diretório do sistema gerado. Veja na seguinte figura.

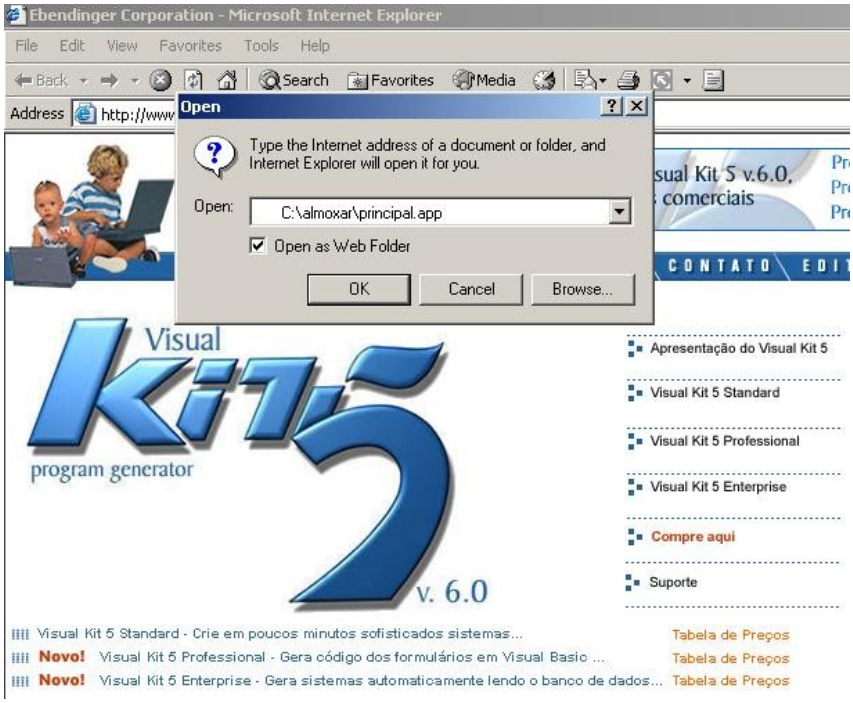


Fig 210 Localizando arquivo principal.app no diretório do sistema gerado.

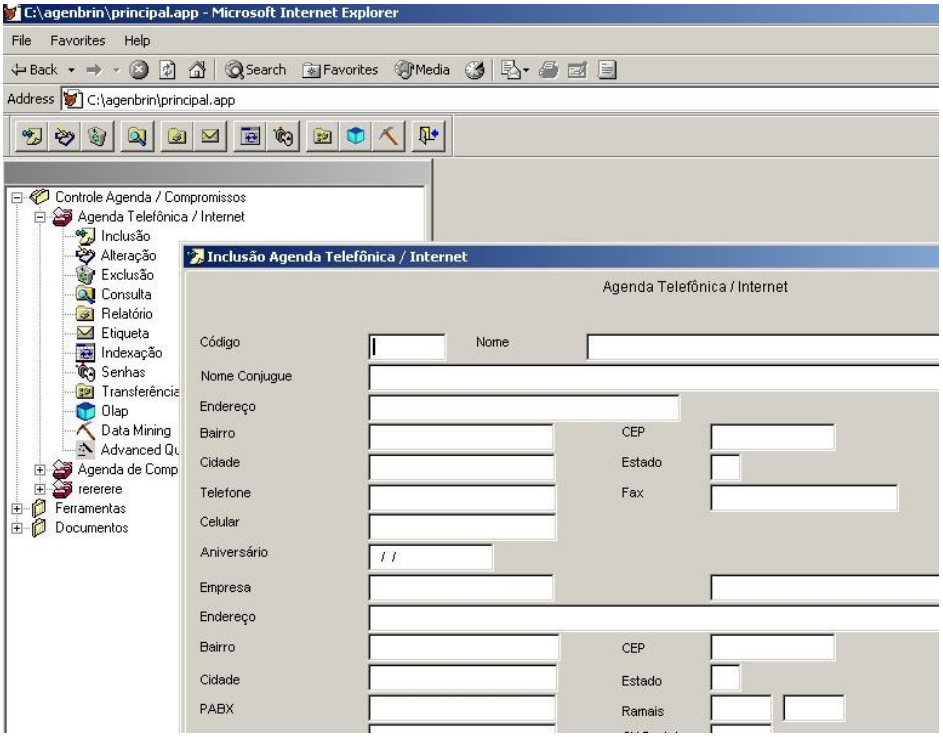


Fig 211 O sistema rodando no Internet Explorer.

XVII. Conhecendo as funcionalidades dos Sistema Gerados

Quando você cria sistemas usando o Visual Kit 5 ele coloca diversos recursos nos sistemas gerados. Cada recurso tem suas finalidades e funcionalidades que descrevemos a seguir.

Localizar entidades e campos do sistema

Utilize a janela “Localizar” para encontrar dados em todos os arquivos do sistema através de palavras chave sem precisar abrir o arquivo.

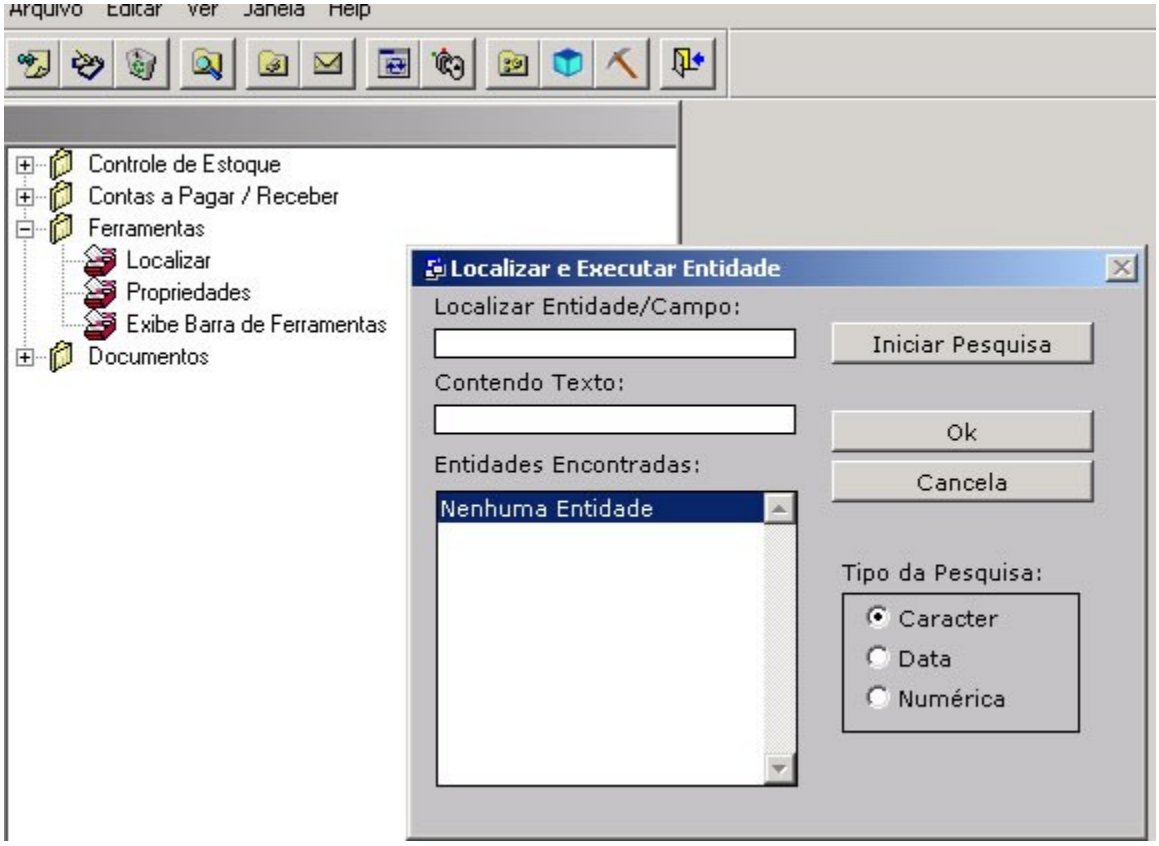


Fig 212 Janela Localizar

Executando o recurso Localizar

Feche todos os arquivos abertos.
Selecione a operação “Consultas”.
Na *treeview* do sistema gerado, abra "Ferramentas" selecione a opção “Localizar”.
Localizando uma entidade (arquivo) ou campo de arquivo
No campo “Localizar” digite a palavra chave inteira ou um trecho. Exemplo: Clientes ou Cli.
Pressione o botão “Iniciar Pesquisa”.
Na lista “Entidades Encontradas” clique sobre a entidade desejada.
Pressione o botão “Ok”.

Localizando uma informação dentro do arquivo

Clique com o mouse no campo “Tipo de Pesquisa”.
No campo “Contendo Texto” digite a palavra chave inteira ou um trecho. Exemplo: SILVA ou LVA.
Pressione o botão “Iniciar Pesquisa”.
Na lista “Entidades Encontradas” clique sobre a entidade desejada.
Pressione o botão “Ok”.
Nos dois casos acima descritos, o sistema executará a operação “Consultas” e posicionará o cursor sobre o primeiro registro encontrado.

Criando usuários (*login* e *senha*) para o sistema

Utilizando a opção **Propriedades** no sistema gerado, você pode cadastrar os usuários que acessarão o sistema com nome (*login)* e senha. Você poderá definir os direitos de cada um, indicando quais módulos e entidades o usuário tem direito incluir, excluir, consultar, etc.

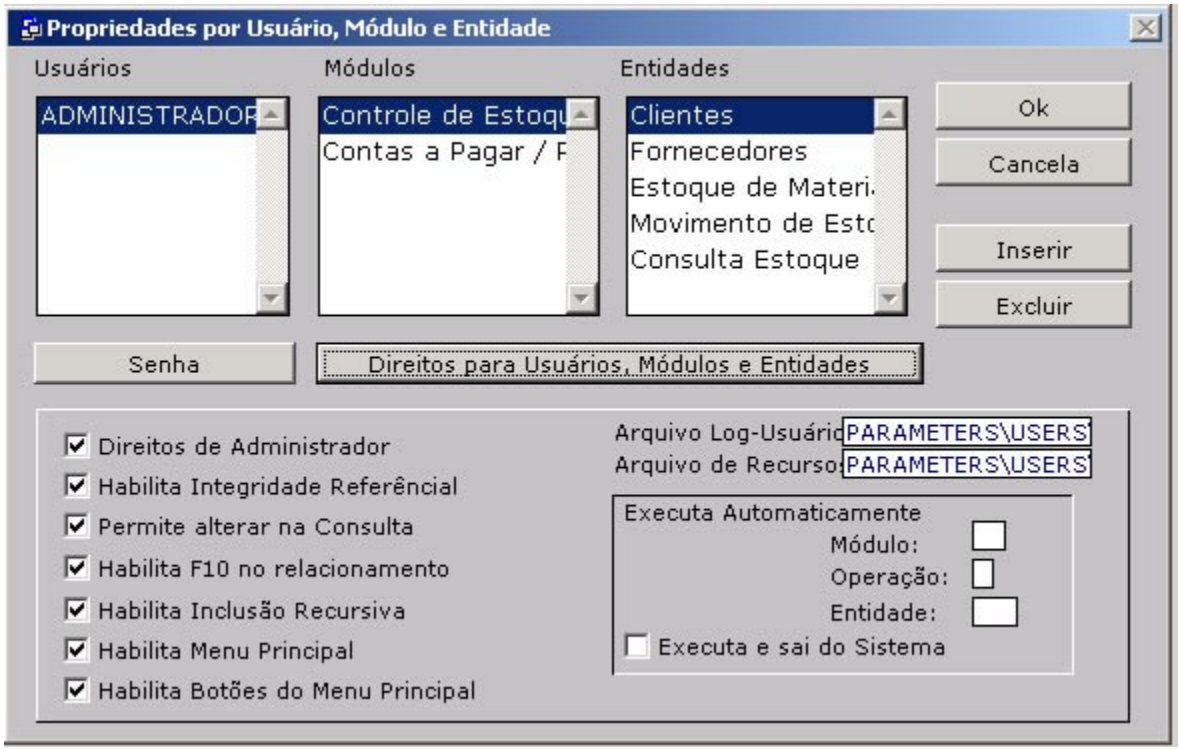


Fig 213 Opção de Propriedades

Quando você entra pela primeira vez na janela “Propriedades” o sistema cria um usuário de nome “**administrador**” com senha igual a “**admini**”. O tamanho da senha é de 6 (seis) caracteres ou dígitos.

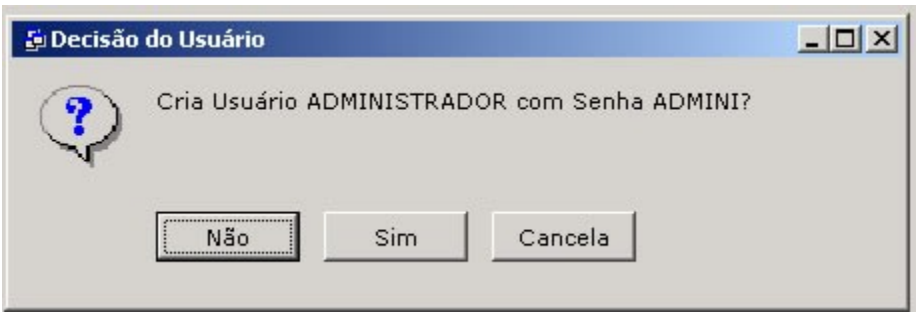


Fig 214 Criando usuário Administrador

Executando as Propriedades
Feche todos os arquivos abertos.
Selecione na treeview do sistema gerado o item "Ferramentas" e a opção “Propriedades”.

Criando usuários
Clique na lista “Usuários”.
Pressione o botão “Inserir”.
Digite o nome do usuário e pressione o botão “Ok”.
Se o sistema gerado vai rodar em rede é obrigatório criar todos os usuários do sistema.

Incluindo / alterando a senha do usuário
Selecione o usuário na lista “Usuários”.
Pressione o botão “Senha”.
Digite a senha do usuário e pressione o botão “Ok”.

Excluíndo usuários
Clique no nome do usuário na lista “Usuários”.
Pressione o botão “Excluir”.
Confirme a exclusão.

Estabelecendo direitos de usuários, módulos e/ou entidades
Selecione o usuário, módulo e/ou entidade na lista correspondente.
Pressione o botão “Direitos para Usuários, Módulos e Entidades”.
Clique uma vez sobre a operação que você quer desabilitar na janela “Direitos”.
Pressione o botão “Ok” para confirmar os dados.

Desabilitando recursos e menus (configura tela e sistema)
Clique sobre o usuário na lista de “Usuários”.

Clique sobre o recurso/menu que você deseja desabilitar para aquele usuário. A seta indica que o recurso/menu está habilitado.

Criando um arquivo de log por usuário
Clique sobre o usuário na lista de “Usuários”.
Clique no campo “Arquivo Log-Usuário” e digite o path e nome do arquivo (TXT) onde serão gravadas as informações da utilização do sistema.

Exemplo: \WINDOWS\RELLOG.TXT.
Criando um arquivo de recursos
Clique sobre o usuário na lista de “Usuários”.
Clique no campo “Arquivo de Recursos” e digite o path e nome do arquivo (*.DBF) onde serão gravadas as informações sobre os recursos do sistema.

Exemplo: \VDEMOS\FOX01.DBF.
Se a execução do sistema gerado for em rede (multiusuário) é obrigatório criar um arquivo de recursos para cada usuário do sistema.

Estabelecendo a execução automática de uma entidade
Clique no campo “Módulo” da janela “Executa Automaticamente” e digite o número do módulo.
No campo “Operação” digite o número da operação “1 - para Inclusão”, “2 - para Alteração” etc.
No campo “Entidade” digite número da entidade que será executada automaticamente.
Para sair após a execução do sistema, clique sobre o campo “Executa e sai do sistema”.
Sai da janela “Propriedades” clicando sempre no botão “Ok”.

Inclusão, alteração e exclusão de registros

Utilize estas operações para incluir, alterar ou excluir dados no sistema. Para selecionar estas operações clique primeiro na entidade desejada. Será aberto um menu tipo *TreeView* com todas as operações registradas para a dada entidade.

Como exemplo, a Fig 215 mostra um sistema *Controle de Estoque*, com a entidade *Clientes* e a operação *Inclusão* selecionada. Clicando nesta operação será aberto o formulário desenhado por você. Por exemplo, veja na Fig 216.

Fig 215 Operação Inclusão selecionada.

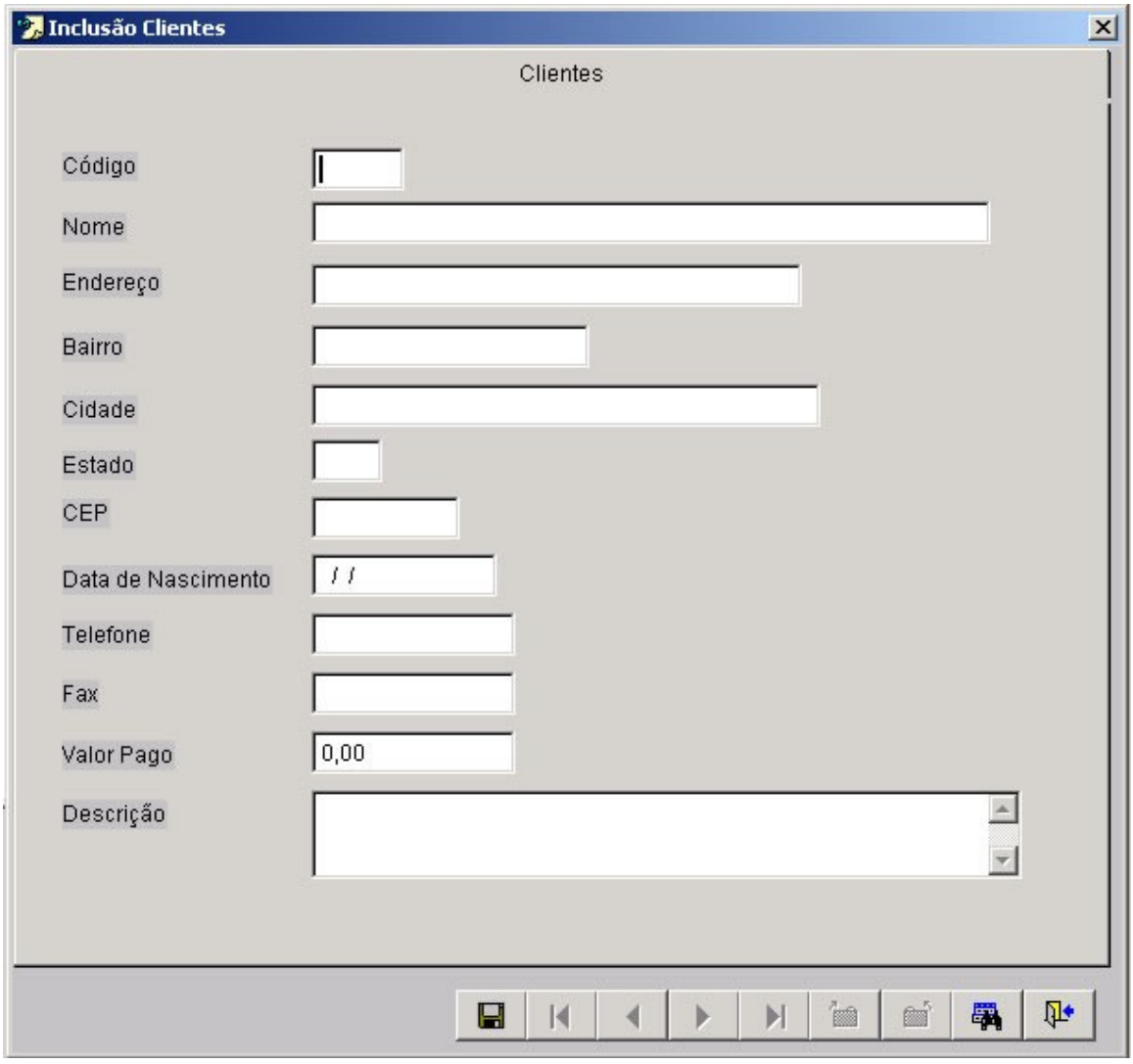


Fig 216 Formulário aberto para a Inclusão.

Selecionada a operação utilize os botões criados pelo próprio Visual Kit 5 apresentados abaixo para: gravar os dados, sair da operação, mudar de registro, consultar o arquivo atual etc.

Grava dados incluídos, alterados.

Na exclusão apaga o registro exibido.

Exibe o primeiro registro do arquivo.

Exibe o registro anterior.

Exibe na tela o próximo registro.

Exibe o último registro do arquivo.

Vai para a tela anterior deste mesmo arquivo.

Vai para a próxima tela deste arquivo. O botão na cor cinza está desabilitado.

Executa a operação de consulta para o arquivo atual.

Cancela a operação.

Para as operações Alteração e Exclusão funcionam de maneira similar.

Dica

Você pode utilizar o mouse ou a tecla <TAB> para movimentar o cursor pelos campos do arquivo. Utilize o mouse também para navegar pelas telas do arquivo.

Consultas

A operação “Consulta” possui todos os recursos necessários para a manipulação completa dos dados de seus arquivos. Esta é uma operação muito útil e usada pelos usuários dos sistemas gerados.

A operação “Consulta” pode ser acessada também através das operações: “Inclusão”, “Alteração”, “Exclusão”, “Relatórios”, “Etiquetas” e “Transferência”.

Nas operações de entrada de dados (“Inclusão”, “Alteração” e “Exclusão”) você pode acessar os dados de arquivos relacionados pressionando a tecla “F10” sobre o campo de relacionamento.

Você pode criar ainda macro entidades de *alteração* e *exclusão,* afim de fazer uma manutenção simplificada de seus arquivos.
Exemplo de consulta ao arquivo (entidade) Clientes.

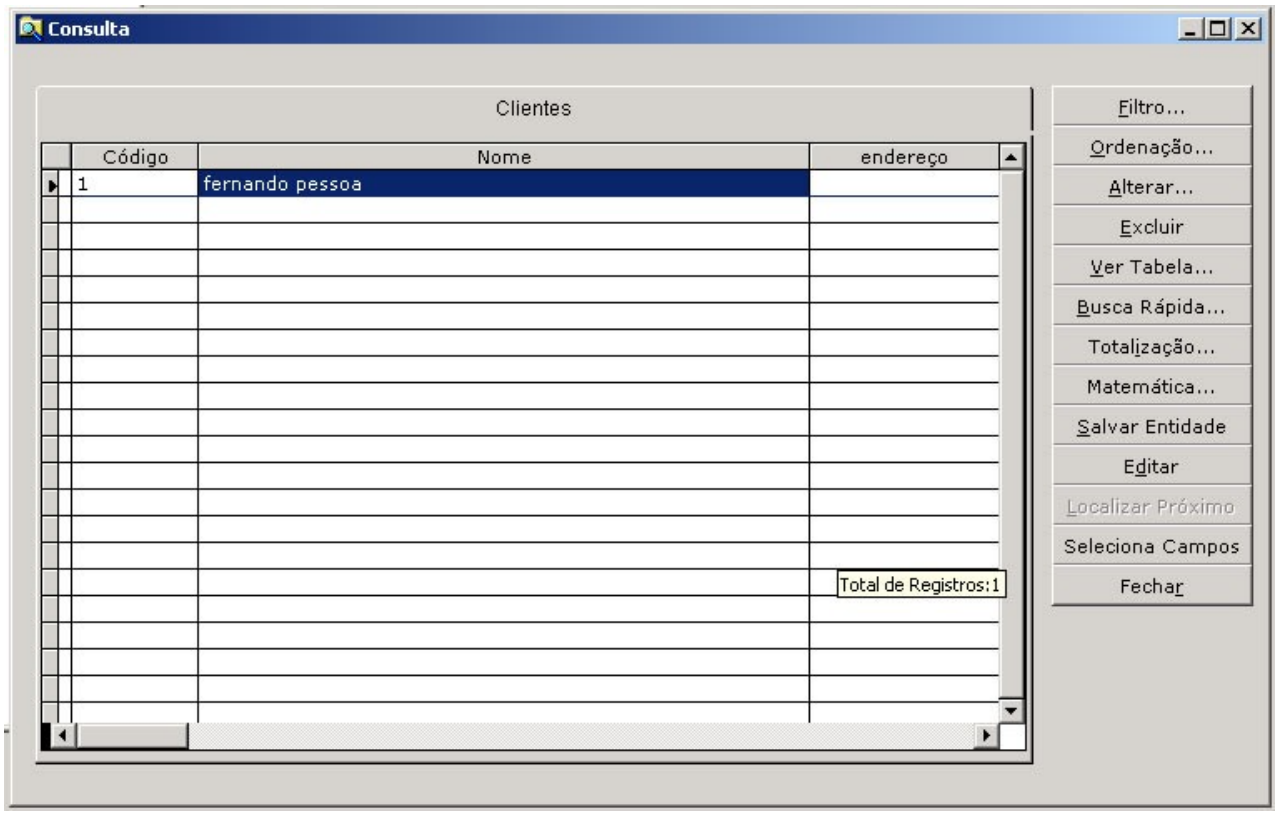
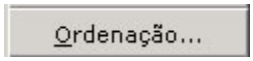


Fig 217 Janela da Operação *Consulta*

Ao selecionar:



Executa a ordenação do arquivo por campos. Descendente ou ascendente.

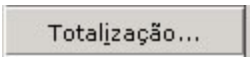
Pesquisa por qualquer campo, inclusive campo *Memo*.

Permite alterar um ou mais registros e salvar macro entidades de alteração.

Permite excluir registros e criar *macro* entidade de exclusão.

Possibilita o acesso aos arquivos relacionados com o arquivo atual.

Pesquisa indexada. Busca uma determinada informação mais rápido que o Filtro.



Calcula o desvio padrão, média aritmética etc, baseado em formula pré-estabelecida.

Cria colunas com o resultado de operações matemáticas.

Salva como *Entidade Macro* de consulta, para automatizar as consultas futuras.

Edita o registro atual através da operação “Alteração” abrindo o formulário da entidade.

Localiza próximo registro que satisfaz a condição estabelecida na janela “Localizar” do menu “Arquivo”. Botão na cor cinza não pode ser utilizado.

Permite alterar os campos que serão exibidos na consulta e sua ordem.

Encerra a operação.

Relatórios rápidos, etiquetas e transferência

O sistema gerado pelo Visual Kit 5 possui todos os recursos para criar relatórios, etiquetas (mala-direta), importar arquivos ou exportar arquivos para outros softwares.

As três operações possuem a mesma interface ou tela. Dependendo da operação selecionada o sistema habilitará alguns botões e desabilitará outros.

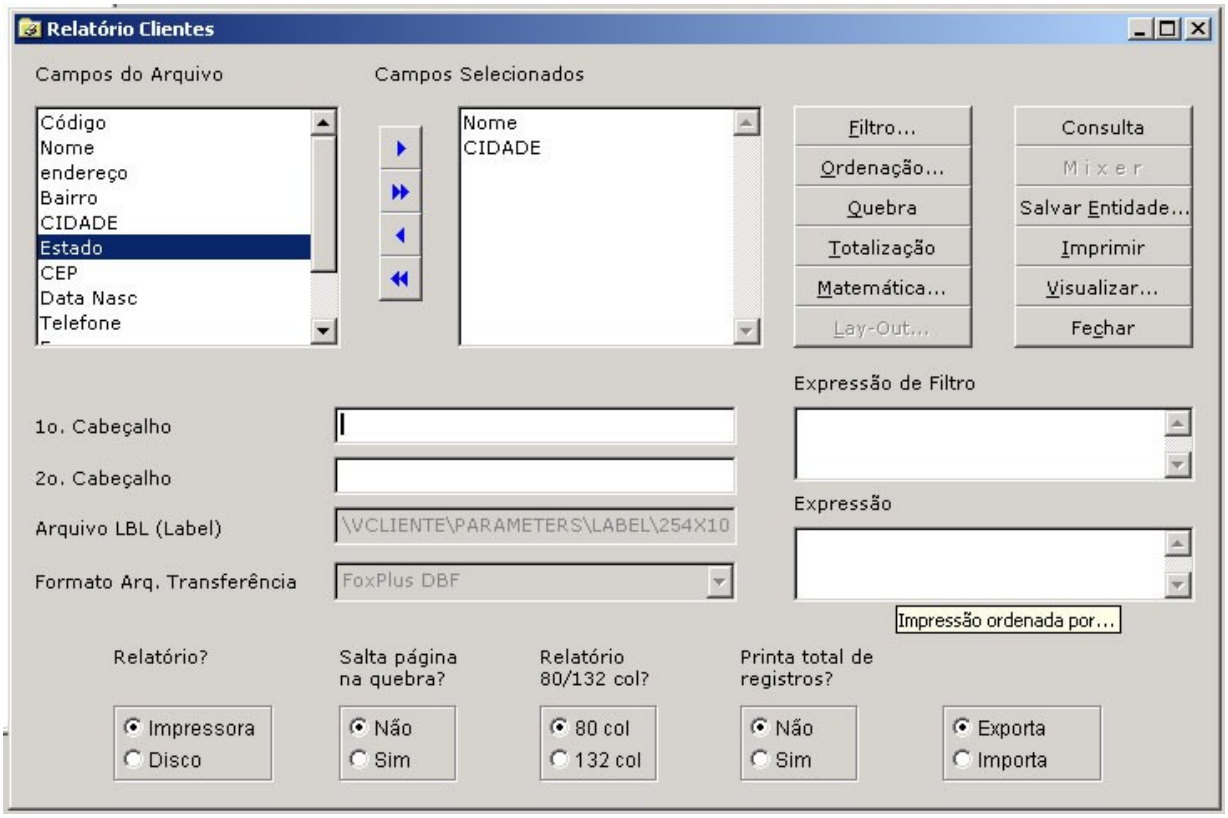


Fig 218 Janela para as operações Relatórios, Etiquetas e transferência de arquivos.

Veja abaixo a descrição das listas e botões desta janela:

Campos do Arquivo:
Lista de campos do arquivo atual e arquivos relacionados. Clique com o mouse no campo que será selecionado.

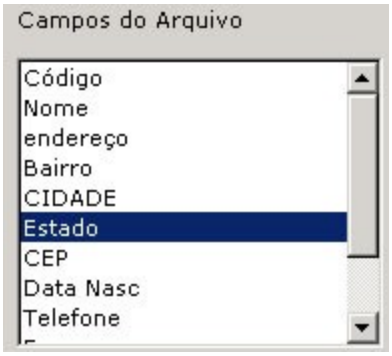


Fig 219 Mostra os campos da entidade selecionada.

Campos Selecionados:
Lista de campos que foram selecionados para impressão ou exportação.

Clique com o mouse no campo para: remover, efetuar quebra, totalização ou criar coluna matemática.

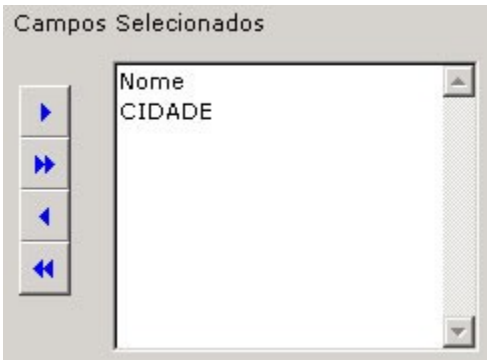


Fig 220 Mostra os campos selecionados.

Seleciona / Remove

Botões para selecionar ou remover campos para impressão ou exportação. Na lista “Campos do Arquivo” clique sobre o campo desejado. Utilize os botões abaixo para selecionar ou remover um ou mais campos.

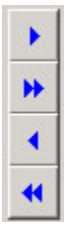
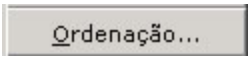


Fig 221 Botões para seleção dos campos

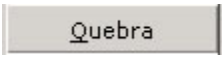
Filtro...

Executa a função “Filtro”. Permite selecionar um grupo de registros para impressão ou exportação.



Ordenação...

Executa a função “Ordenação”. Permite ordenar os registros que serão impressos ou exportados.



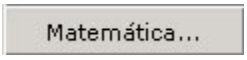
Quebra

Divide a impressão por grupos.
Na lista “Campos Selecionados” clique sobre o campo que formará a divisão. Exemplo: divisão da impressão por “Estado”.
Pressione o botão “Quebra”

Totalização

Totaliza campo numérico. Se existir quebra, sub-totaliza a cada novo grupo.

Na lista “Campos Selecionados” clique sobre o campo que será totalizado.
Exemplo: imprimir o total de salários ao final do relatório.
Pressione o botão “Totalização”

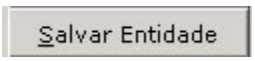


Matemática...
Executa a função “Matemática”. Cria colunas matemáticas com o resultado da expressão matemática estabelecida.
Na lista “Campos Selecionados” clique sobre o campo que será calculado.
Exemplo: imprimir coluna, calculando 10% de desconto no preço de venda.
Pressione o botão “Matemática”

Lay-Out...
Opção desabilitada.
Atenção A quebra não tem efeito sobre a impressão do Lay-Out. O filtro e a ordenação sim.

Consulta
Exibe os dados do arquivo atual, através da operação “Consulta”.

Mixer Opção desabilitada.



Salvar Entidade...
Salva macro entidade. Monta um relatório, etiqueta, importação ou exportação de dados.
Pressione o botão “Salvar Entidades” para criar uma nova entidade no menu de entidades.
Sempre que selecionada esta entidade, o sistema irá executar a impressão do relatório ou etiqueta, importar ou exportar os dados automaticamente.

Imprimir / Processar
Executa a impressão do relatório ou etiqueta, importa ou exporta os dados.

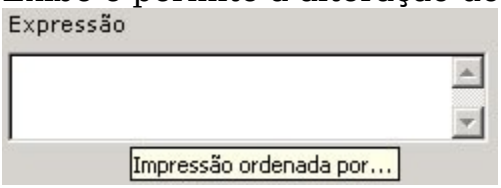
Visualizar...
Exibe na tela arquivos textos criados no disco (winchester).

Fechar
Encerra a operação.

Expressão de Filtro
Exibe e permite a alteração do filtro (condição de pesquisa) ativo.

Indexador Ativo (Expressão)

Exibe e permite a alteração do ordenador ativo.

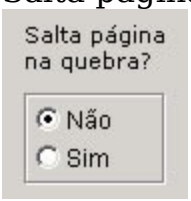


1o. Cabeçalho
Primeiro cabeçalho do relatório.

2o. Cabeçalho
Segundo cabeçalho do relatório.

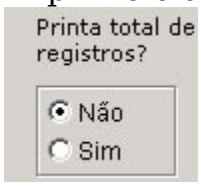
Relatório? Impressora / Disco
Envia o arquivo para impressão ou exportação para disco ou impressora ativa.

Salta página na quebra? Não / Sim
Salta página a cada mudança de grupo quando existir quebra.

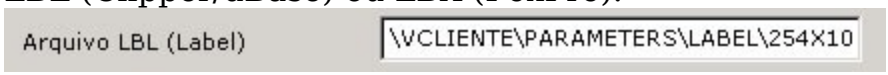


Relatório 80/132 col? 80 col / 132 col
Impressão em formulário de 80 ou 132 colunas.

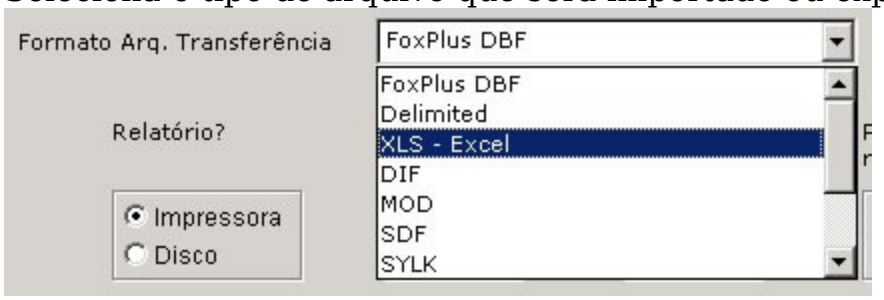
Printa total de registros? Não / Sim
Imprime o total de registros impressos ao final do relatório.



Arquivo LBL/LBX (label)
Nome do arquivo de etiqueta que será utilizado. Arquivo com terminação LBL (Clipper/dBase) ou LBX (FoxPro).



Formato Arq. Transferência
Seleciona o tipo de arquivo que será importado ou exportado.



Arquivo Descrição
FOXPLUS Arquivo de dados FoxBase+, dBase, Clipper, Foxpro.
DELIMITED Arquivo ASCII (tipo texto).
Excel - XLS Planilha Excel versão 2.x ou superior.
DIF Planilha VisiCalc 1.0.
MOD Multiplan.
SDF Arquivo ASCII (tipo texto).
SYLK Formato de intercâmbio de ligação SYMBOLIC usado pelo Microsoft Multiplan.
WK1 Lotus 1-2-3 versão 2.x.
WKS Lotus 1-2-3 versão 1A.
WR1 Planilha Symphony versão 1.1.

WRK Planilha Symphony versão 1.01.

Exporta / Importa
Ativa importação ou exportação na transferência.

Senhas dentro da entidade

Esta função permite a inclusão de senhas de acesso às entidades por operação básica. Você pode criar uma senha para cada operação por entidade.
Se a entidade possuir senha de acesso, sempre que selecionada, o sistema exigirá a entrada do nome do usuário e senha.

Você tem três chances para acertar a senha, caso contrário não será possível acessar a entidade.

Procedimentos para criar senhas por entidade

- Utilizando o mouse selecione a operação “Senhas” no menu treeview do sistema.
- Selecione com o mouse a operação que receberá a senha, digite a senhas (de 1 até 6 caracteres ou dígitos) utilizando o mouse pressione o botão “Salvar” para gravar as senhas ou “Cancela” para abandonar.

Dica:
Se você deseja evitar o acesso a operação “Senhas”.
- Desabilite a operação “Senhas” para cada entidade do sistema.
- Ou crie uma senha para a operação “Senhas” por entidade.

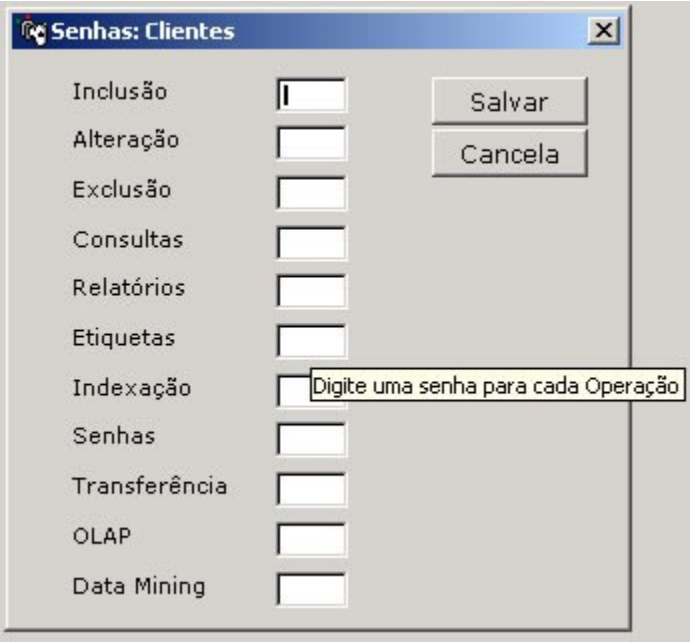


Fig 222 Colocando senhas nas operações das entidades

Indexação

Quando você gera sistemas em formato .DBF, e você exclui regularmente informações dos seus arquivos, execute esta operação para apagar fisicamente todos os registros excluídos. Dessa forma, os arquivos se manterão sempre com o número real de registros.

Atenção!

Se você alterar a estrutura de uma entidade, ou seja, incluiu ou excluiu novos campos no formulário, mudar o tipo de dados de um campo, etc. indexe a entidade antes de usá-la. Usar uma entidade que teve sua estrutura alterada antes de indexá-lo causará problemas.

Procedimentos para indexar uma entidade.

- Utilizando o mouse selecione a operação “Indexação” na treeview do sistema.
- O sistema inicialmente fará a exclusão física dos registros excluídos, em seguida indexará o arquivo.
- O path (*caminho*) e nome dos arquivos de dados, de índice e a chave de indexação também serão exibidos nesta tela.

Atenção! Não interrompa esta operação

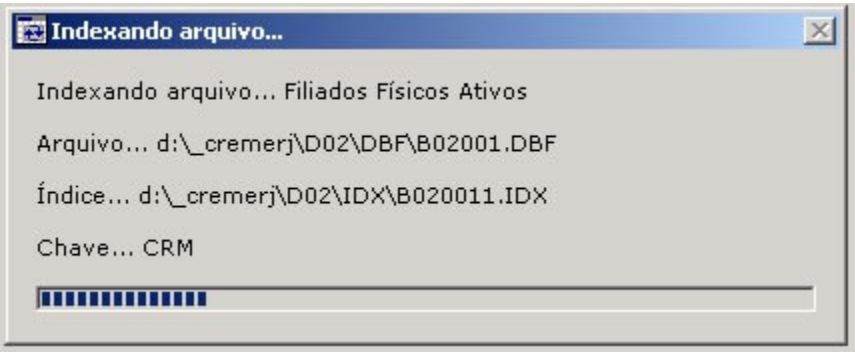


Fig 223 Indexando as tabelas das entidades

Em ambiente de rede local, certifique-se de que o arquivo a ser indexado não está sendo utilizado por nenhum usuário da rede. Se você tentar indexar um arquivo que está sendo utilizado, o sistema exibirá uma mensagem de erro, não indexará o arquivo selecionado e poderá perder os dados do arquivo.

Outras opções úteis

Filtro e ordenação

Filtro
Seleciona em um arquivo com (n) registros somente um grupo de registros.
Por exemplo: em um arquivo com mil estabelecimentos de ensino, utilizando o filtro você pode selecionar somente os estabelecimentos do *bairro igual a Catete* ou os *nomes que começam pela letra A*.

Esta função permite relacionar mais de um campo para filtro através dos operadores lógicos “E” e “OU”.

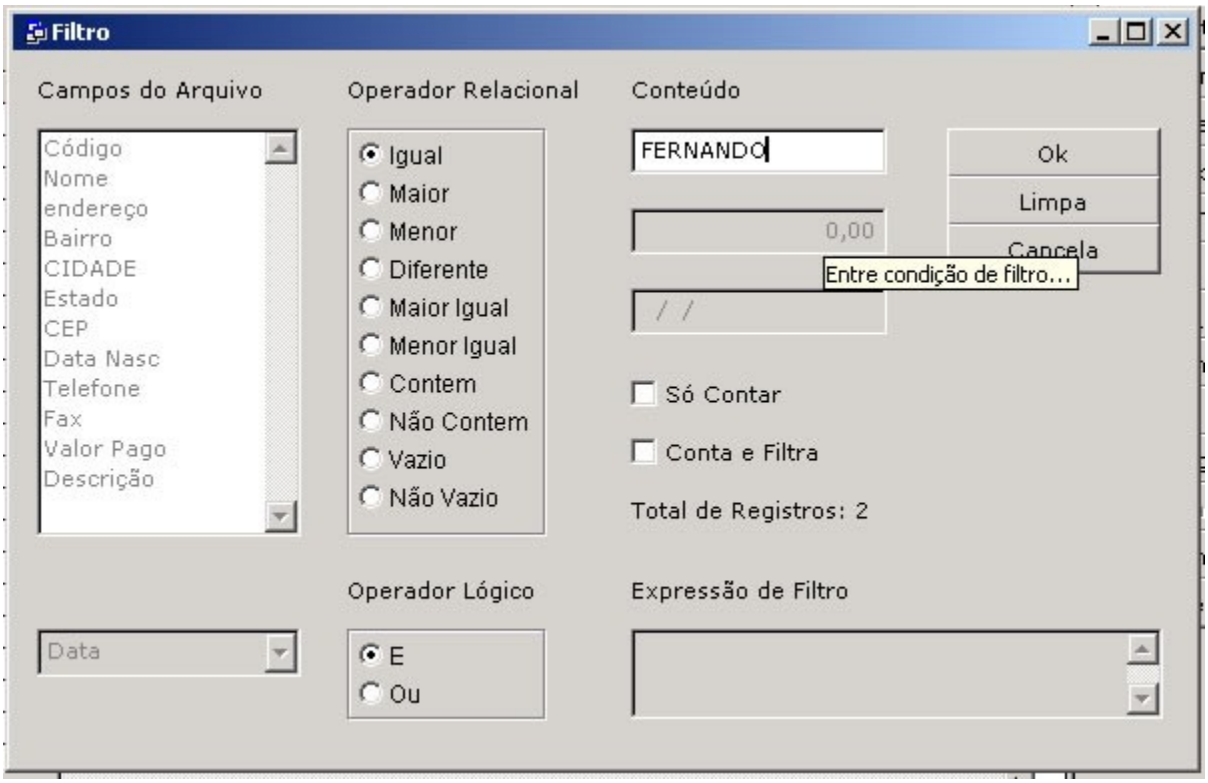


Fig 224 Janela de Filtro

Para estabelecer uma condição de filtro (pesquisa)
Você pode estabelecer condições de filtro para destacar de um universo "X" qualquer uma determinada porção de registros.
Como por exemplo:
Campo NOME que começa pela letra *A (ana, alberto...)*;
Campo DATA com dia igual ou maior de *10 (10, 12,20...)*;
Campo ENDEREÇO que contém a string *Senhora (Av. Nossa Senhora...)*;
Campo VALOR_PAGO igual a *vazio (0)*;
Campo NOME contém *Silva (Ernesto da Silva...)*;
Campo CEP menor igual a *20000 (19000,18000...)*;
Campo NASCIMENTO com ano igual a *1966*.

Filtrando um único campo (caractere ou numérico)
O filtro pode ser utilizado nas operações consulta, relatório, etiqueta e transferência e sua manipulação é sempre idêntica.
O exemplo abaixo é genérico e pode ser utilizado em qualquer uma das operações citadas.
Utilizando o mouse pressione o botão “Filtro”.
Selecione, na caixa de texto, o campo que será filtrado, pressione o botão do mouse duas vezes sobre o campo ou tecle “ENTER”.
Selecione o operador relacional.
Digite o conteúdo que será filtrado.
Utilizando o mouse pressione o botão “Ok”.

Filtrando um campo data
Utilizando o mouse pressione o botão “Filtro” ou selecione “Filtro” no menu “Funções”.
Selecione, na caixa de texto, o campo do tipo *DATA* que será filtrado, pressione o botão do mouse duas vezes sobre o campo ou tecle “ENTER”.
Selecione a opção de data a filtrar.
Selecione o operador relacional.
Digite o conteúdo que será filtrado.
Utilizando o mouse pressione o botão “Ok”.

Filtrando mais de um campo
Execute os passos acima para selecionar cada campo.
Após digitar o conteúdo do campo tecle “ENTER”;
Em seguida o cursor cairá automaticamente na caixa de operadores Lógicos “E” e “OU”, selecione o botão de rádio desejado e repita a operação descrita na opção filtrando um campo.
Para finalizar pressione o botão “Ok”.

Ordenação
Seleciona os campos, que farão parte da nova chave de ordenação do arquivo, permitindo a visualização do arquivo ordenado ascendente ou descendente por outros campos além dos campos que fazem parte da chave primária do arquivo.

Por exemplo: em um arquivo que possui os campos *código, nome e endereço*, você pode ordenar a sua visualização na tela ou na impressora por endereço em ordem *ascendente*, isto é, da letra “A” para a letra “Z” ou descendente (de “Z” para “A”).

Fig 225

Para ordenar o arquivo
Ordenar o arquivo pelo campo NOME do cliente *(de A até Z ou Z até A)*;
Ordenar o arquivo pelo campo DATA de nascimento *(da menor para a maior data)*;
Ordenar o arquivo pelo campo SALÁRIO *(do maior para o menor)*.

Ordenando um único campo do arquivo
Utilizando o mouse pressione o botão “Ordenação”.
Selecione, na caixa de texto, o campo que será ordenado, pressione o botão do mouse duas vezes sobre o campo ou tecle “ENTER”.
Selecione o indexador ascendente ou descendente.
Utilizando o mouse pressione o botão “Ok”.

Ordenando mais de um campo
Execute todos os passos descritos acima.
Após selecionar o primeiro campo selecione o próximo campo que será ordenado e repita a operação descrita na opção “Ordenando um único campo”.
Para finalizar pressione o botão “Ok”.

Quebra.

Divide a impressão das informações por grupos (quebra), para facilitar a visualização dos relatórios.
Por exemplo: em um arquivo que possui os campos *Número da RA (Região Administrativa), Código do bairro, Nome da Empresa e Endereço completo,* você pode imprimir as empresas agrupadas por "*RA*", quebrando por "*RA*".
O sistema fará a impressão de cada "*RA*" uma única vez e irá saltar uma linha ou página a cada mudança de "*RA*".

Para dividir a informação impressa por grupos
Você pode dividir a informação impressa por grupos, como por exemplo:
Quebrar a impressão pelo campo DATA de inscrição *(imprime por grupo os dados dos registros que tem campo DATA de inscrição igual)*;
Quebrar a impressão pelo campo CEP *(imprime grupos de informação com o CEP idêntico);*
Quebrar por Cidade *(imprime cidades iguais agrupadas)*.

Quebrando por um único campo
Utilizando o mouse selecione (pressione) o campo que será quebrado na caixa de "Campos Selecionados".
Pressione o botão "Quebra".

Quebrando por mais de um campo
Repita a operação acima para todos os campos que você deseja quebrar.
Continue repetindo os passos descritos acima para a escolha de todos os campos da quebra.
Importante Os campos selecionados na quebra devem ser os primeiros campos escolhidos na caixa de texto "Campos Selecionados" e não pode haver nenhum campo entre eles que não faça parte da quebra.
O *indexador ativo* deve ser formado pelos campos que fazem parte da quebra

---

Operações Matemáticas

Cria colunas (virtuais) com o resultado de expressões matemáticas pré-estabelecidas.
Esta função permite *somar*, *subtrair*, *multiplicar*, *dividir, tirar o percentual* ou totalizar campos numéricos, criando colunas virtuais.

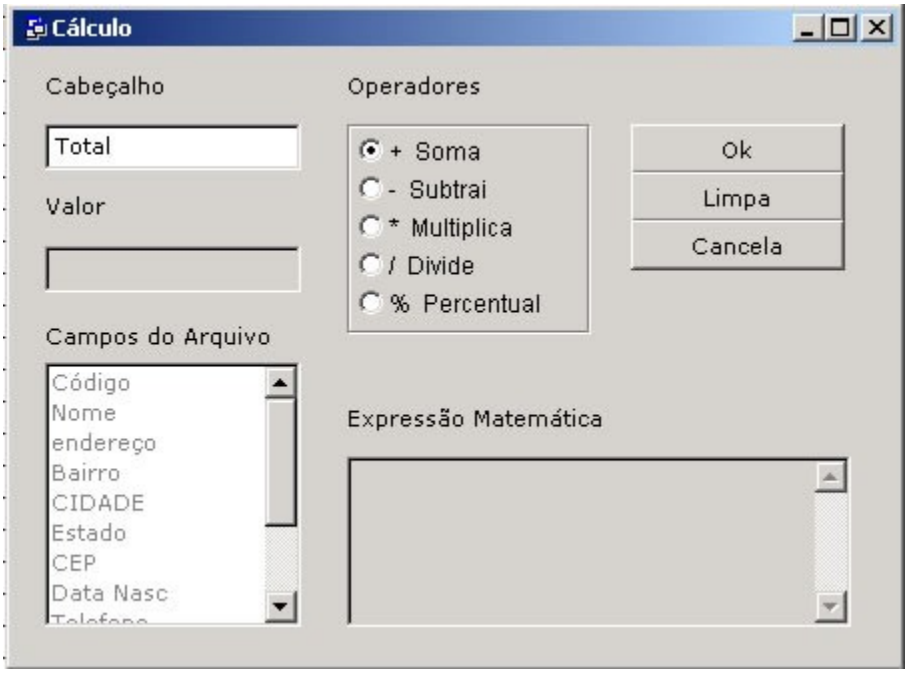


Fig 226

Para totalizar uma coluna numérica na consulta
Para totalizar um campo numérico, selecione a opção “Totalização”.
Digite o nome do cabeçalho de totalização;
Escolha o campo (numérico) que você deseja totalizar;
Selecione o operador matemático;
Digite um valor numérico ou tecle “ENTER” e selecione outro campo;
Ao final pressione o botão “Ok”.
O sistema fornecerá automaticamente os seguintes resultados:
Total de registros (total de registros calculados);
Soma dos valores (soma da expressão matemática registro a registro);
Média (média aritmética);
Valor mínimo (menor valor calculado encontrado no arquivo);
Valor máximo (maior valor calculado encontrado no arquivo);
Variância da População;
Desvio Padrão.
Para retornar a tela da consulta pressione qualquer tecla.
Importante A multiplicação da coluna por "1" tem por objetivo efetuar a totalização direta de uma coluna numérica.

Para totalizar uma coluna numérica no relatório
O total geral da soma dos campos será impresso ao final do relatório.
Se existir uma quebra o subtotal será impresso a cada mudança de grupo (quebra) e o total geral ao final do relatório.

Para totalizar um campo numérico selecione o campo que será totalizado na caixa de texto “Campos Selecionados” e utilizando o mouse pressione o botão “Totalização”.
Para retirar um campo da totalização remova o campo da caixa de texto “Campos Selecionados”.
Observação A totalização só tem efeito sobre a coluna. Não é possível totalizar as linhas de um relatório.
Não é possível totalizar o resultado das totalizações.

Para criar colunas numéricas calculadas
Utilizando o mouse pressione o botão “Matemática”.
Digite o nome do cabeçalho da coluna matemática;
Escolha o campo (numérico) que você deseja utilizar no cálculo;
Selecione o operador matemático;
Digite um valor numérico ou tecle “ENTER” e selecione outro campo;
Ao final pressione o botão “Ok”.
O sistema criará novas colunas, com o resultado da expressão matemática que você montou.
Alerta A *divisão por ZERO* causará um erro de execução. Filtre os valores diferentes de zero antes de executar uma divisão.

Consultas, Relatórios e Etiquetas Avançadas

A operação Consultas, Relatórios e Etiquetas Avançadas é uma magnífica ferramenta que permite a criação de inúmeras consultas e relatórios a partir das entidades atual, incluindo também dados das entidades relacionadas.

Quando você usa esta operação, é aberta a janela **Gerador de Consultas Avançadas**, como mostra a Fig 227. Esta janela contém diversas páginas que descrevemos a seguir.

Origem dos Dados : preparando os dados

Na página **Origem de Dados**, você pode escolher os campos que deseja consultar, filtrá-os, selecionar as quantidades de registros, etc., ou seja, prepara todo um ambiente de dados que depois será usado para gerar as consultas, relatórios, etiquetas e exportação de dados.

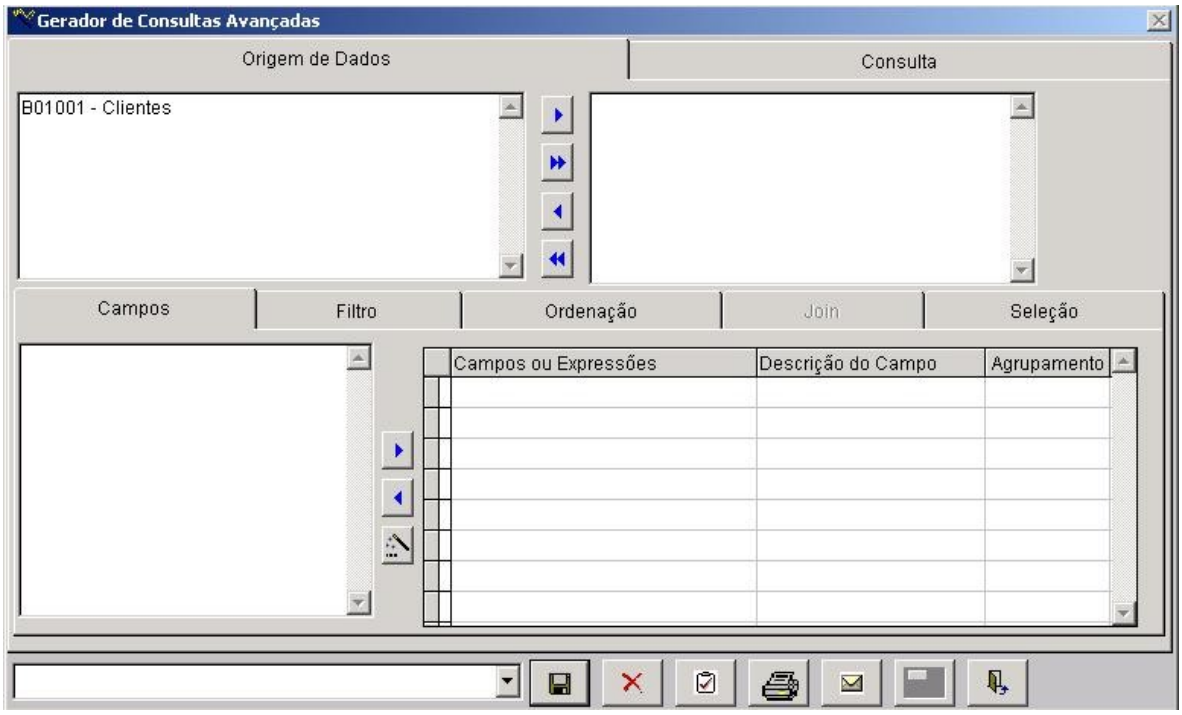


Fig 227 Janela do Gerador de Consultas Avançadas

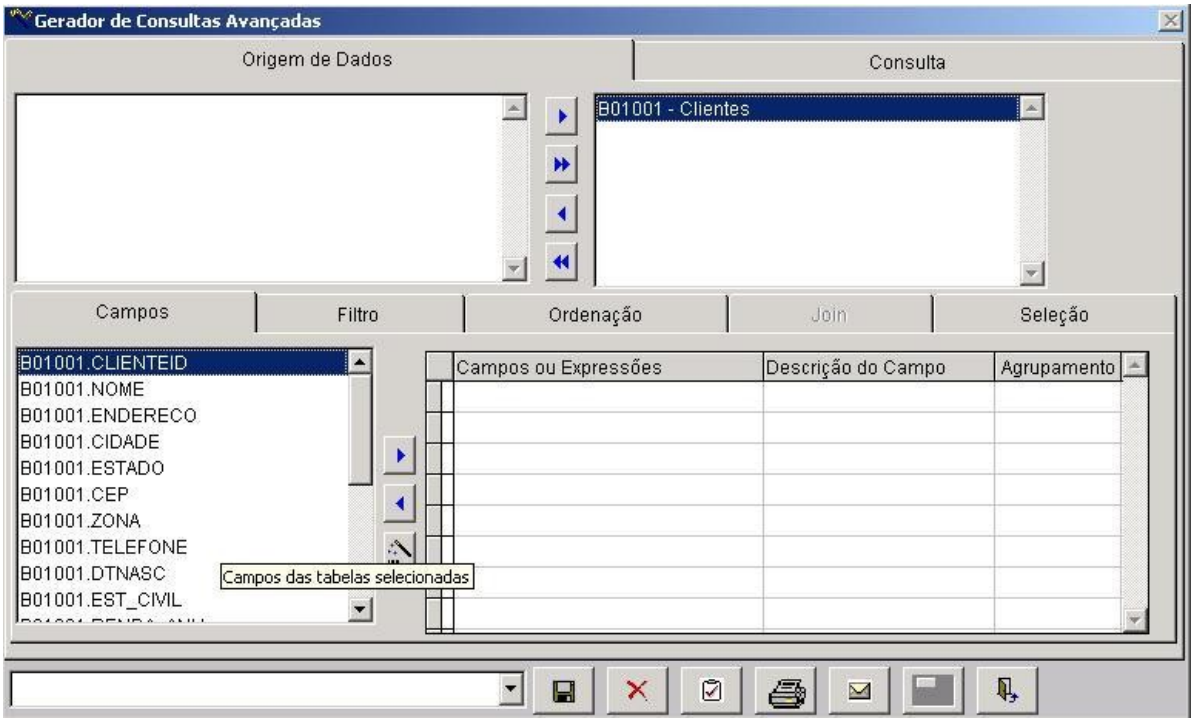


Fig 228 Escolhendo a Tabela

Na parte superior da página "Origem dos Dados" você identifica as entidades ou tabelas, que serão usadas na criação da sua consulta. Você pode usar os botões no meio das duas listas para passar campos de uma lista para outra.

Na parte inferior da janela estão disponíveis 5(cinco) sub páginas: “Campos”-selecionar campos, “Filtro”-filtra os registros, “Ordenação”-ordenar registros, “Join”(Junção)-relacionar tabelas entre si e “Seleção”-seleciona registros.

Atenção A opção **Join** não está disponível nesta versão.

Na sub-página “Campos” são listados todos os campos disponíveis para a criação da sua consulta avançada. Selecione aqui os campos que você deseja apresentar da seguinte forma.

- Faça um clique sobre algum campo da lista **Campos**
- Clique no botão que indica a direção [>]. O campo agora aparece na lista da esquerda. Veja nesta figura:

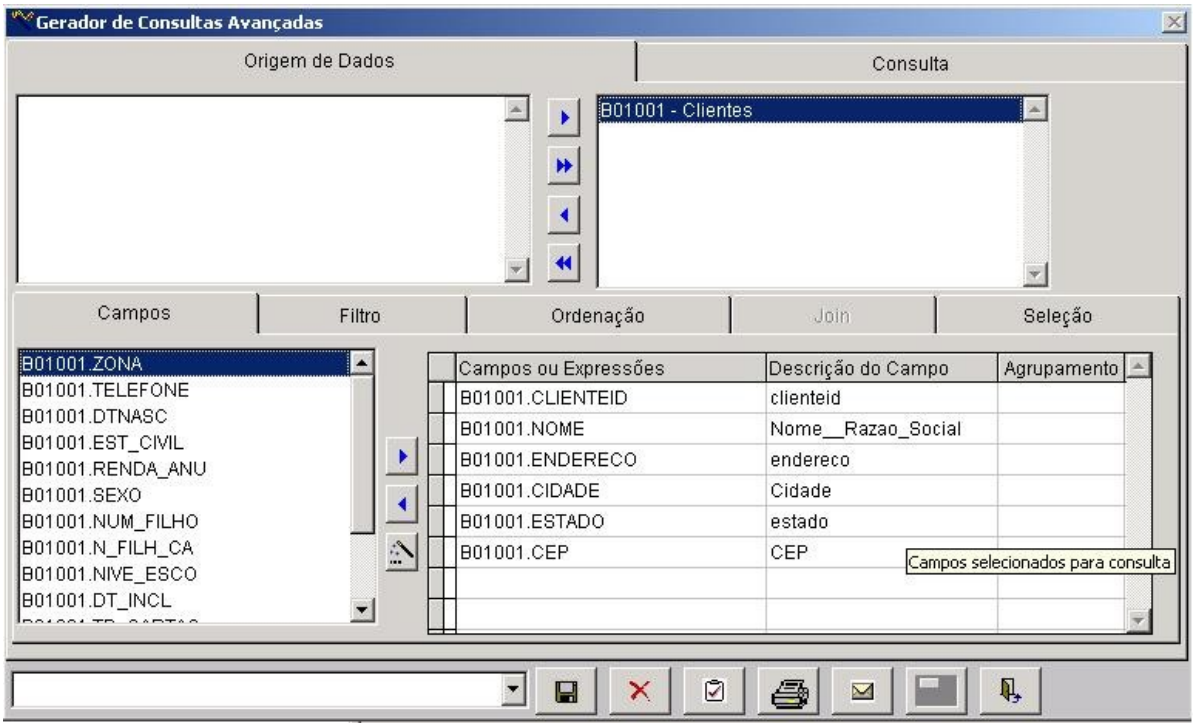


Fig 229 Escolhendo os Campos

Clicando no botão você abre o **Construtor de Expressões** mostrado na Fig 230 pora a criação de expressões de diversos tipos, exemplo expresões matemáticas, de caractere, etc.

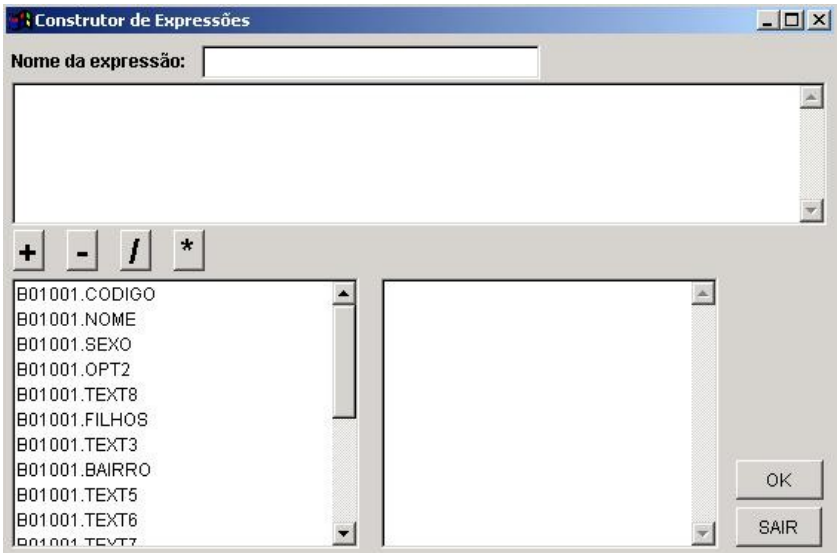


Fig 230 Construtor de Expressões.

A sub-página “Filtro” permite especificar várias condições para selecionar os registros das tabelas. Cada condição pode ser separada pelos operadores lógicos **OU** e **E** que são definidos na coluna **Lógico.**

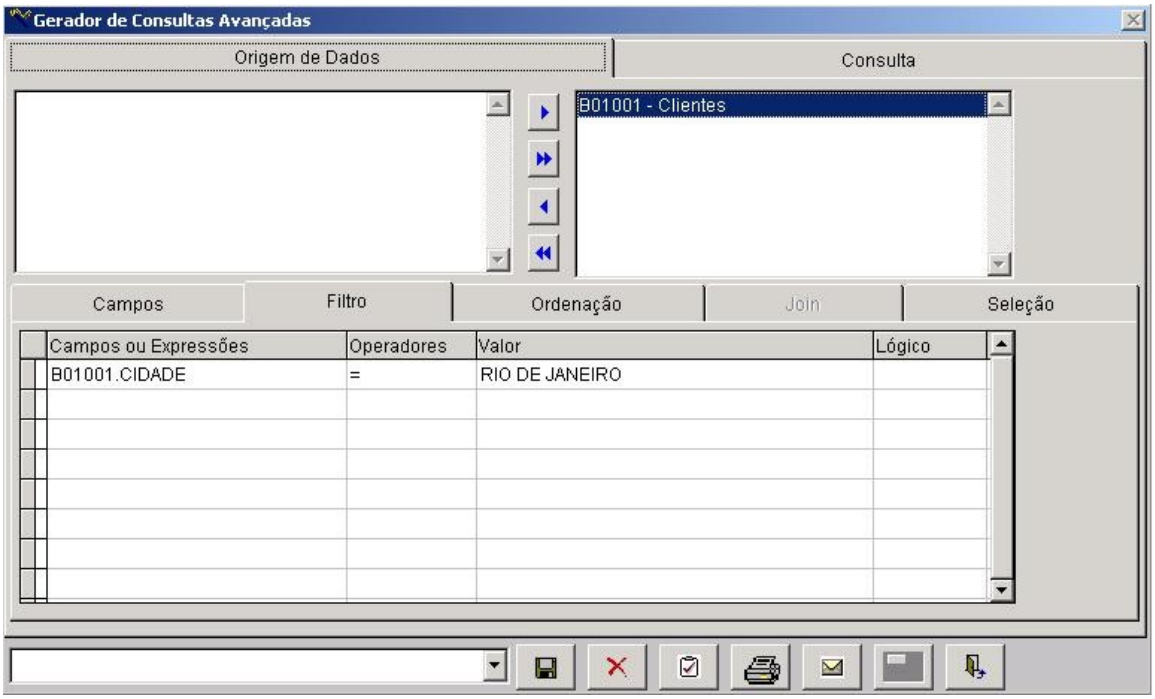


Fig 231 Janela que define o filtro

Ao criar filtros, você estará realmente montando atraves dos operadores uma equação entre a coluna “**Campos ou Expressões”** e o “**Valor”**. Os operadores relacionais disponíveis para criar filtros são os seguintes:

Operador	**Descrição**
= (Igual):	Determina que o valor do Campo seja igual ao Valor. Ex: Campo Físico-Jurídico seja igual a J (Jurídico). Veja na figura acima.
>(maior que):	Especifica que o valor do campo deve ser maior do que o Valor.
>=(maior igual que):	Especifica que o valor do campo deve ser maior ou igual do que o Valor.
<(menor que):	Especifica que o valor do campo deve ser menor do que o Valor.
<=(menor igual que):	Especifica que o valor do campo deve ser menor ou igual do que o Valor.
<>(diferente):	Especifica que o valor do campo deve ser diferente do que o Valor.
Like (contém):	Especifica uma comparação parcial. Por exemplo Campo.Nome like MAR, retornará: MARIA, MAURICIO, MARTA, MARCOS, etc.

A sub-página “**Ordenação**” permite a definição de ordenação dos dados que aparecerão na consulta.

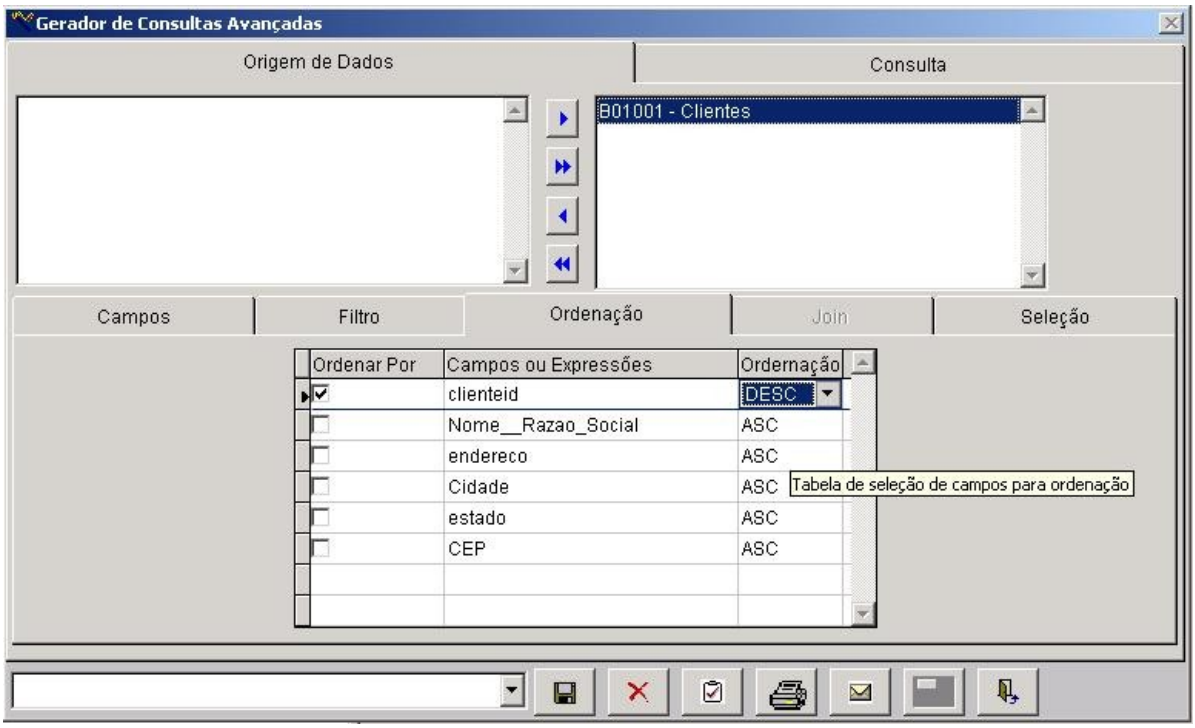


Fig 232 Janela que define o tipo de ordenação

Clicando na coluna "**Ordenar Por**:" você define a ordem em que os registros serão exibidos na consulta ou relatório. Selecione "**ASC**" para ascendente e "**DESC**" para descendente.

Atenção

A ordem de cliques na coluna "**Ordenar Por**:" indica a seqüência da Ordenação.

A sub-página "Seleção" permite definir a quantidade de registros que serão apresentados na consulta, além de disponibilizar o código de seleção para edição.

A opção "Definir a quantidade de registros que serão Visualizados:" habilita o campo de quantidade de registros que serão apresentados no resultado da consulta. O Valor inicial é de 10 registros.

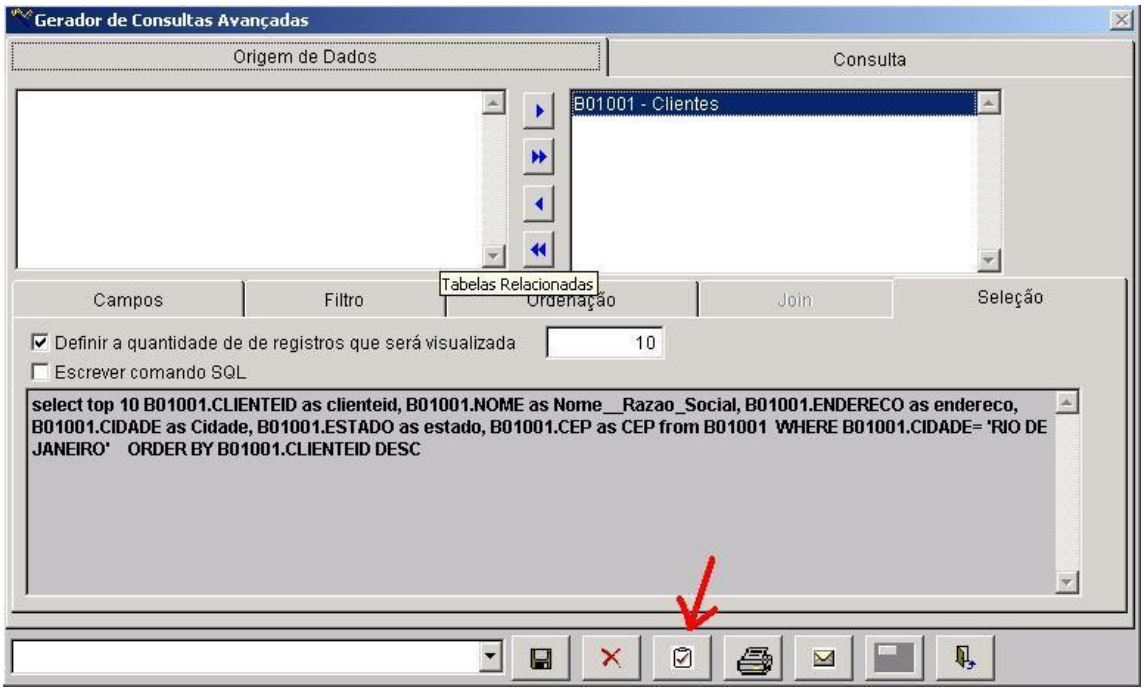
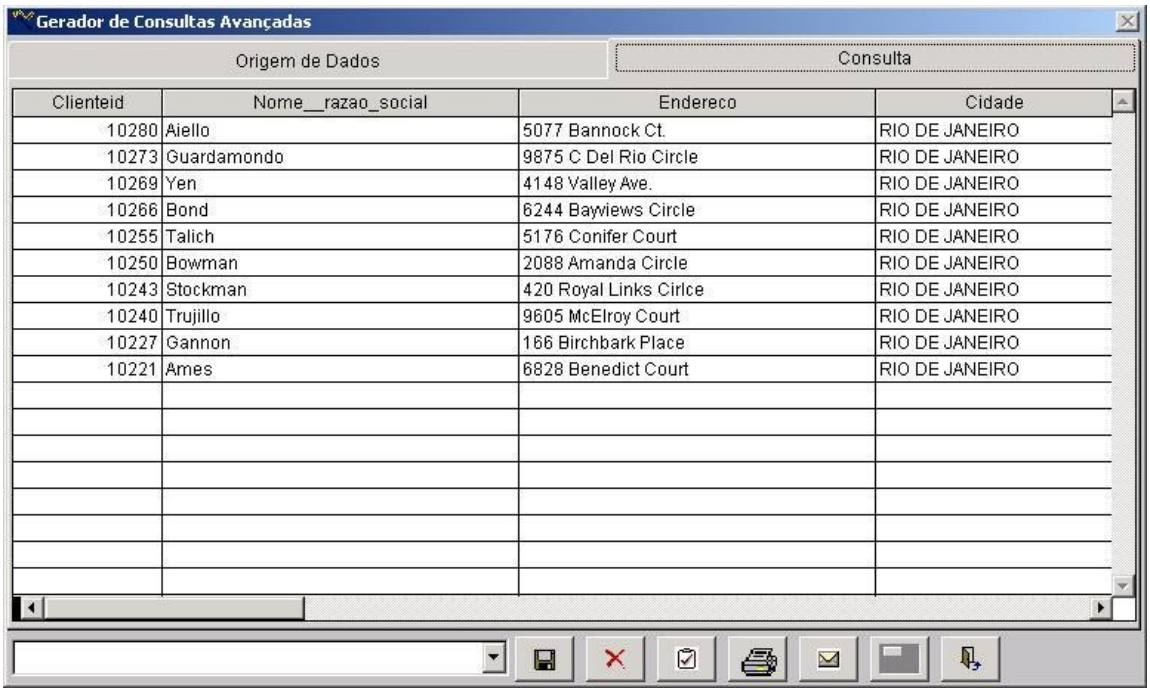


Fig 233 Janela Seleção: Selecionando a quantidade de registros

---

Atenção

Antes de executar um Relatório ou um Etiqueta é preciso primeiramente executar a Consulta.



Clienteid	Nome__razao_social	Endereco	Cidade
10280	Aiello	5077 Bannock Ct.	RIO DE JANEIRO
10273	Guardamondo	9875 C Del Rio Circle	RIO DE JANEIRO
10269	Yen	4148 Valley Ave.	RIO DE JANEIRO
10266	Bond	6244 Bayviews Circle	RIO DE JANEIRO
10255	Talich	5176 Conifer Court	RIO DE JANEIRO
10250	Bowman	2088 Amanda Circle	RIO DE JANEIRO
10243	Stockman	420 Royal Links Cirlce	RIO DE JANEIRO
10240	Trujillo	9605 McElroy Court	RIO DE JANEIRO
10227	Gannon	166 Birchbark Place	RIO DE JANEIRO
10221	Ames	6828 Benedict Court	RIO DE JANEIRO

Fig 234 Resultado da Consulta

Em qualquer momento, durante a montagem da consulta, você pode examinar a declaração SQL que corresponde as suas definições até aquele ponto. É uma maneira bastante rápida de aprender sobre a sintaxe – e poder – do comando SQL. Para habilitar a declaração SQL, use a opção "Escrever comando SQL" habilita a edição do código do **select-SQL**.

Depois de selecionados os campos, filtrados, ordenados, etc. chegou o momento de selecionar a saída dos resultados. Para especificar a saída use os seguintes botões.

Funcionalidade dos botões:

Armazenar uma consulta. Digite o nome da consulta que você deseja gravar e clique no botão Você pode salvar até 999 consultas por entidade. As consultas armazenadas poderão ser alteradas sempre que desejado.

Exclui uma consulta armazenada. Selecione na *Combobox* das consultas gravadas a consulta que você deseja excluir e clique no botão excluir.

Executa a consulta. Uma vez que você tenha definido uma consulta, você poderá executar a consulta e os resultados serão exibidos na página Consulta.

Manda a seleção para um Relatório. As consultas criadas são a base para a criação de sofisticados relatórios. Clique neste botão após a confecção, na seleção de uma consulta armazenada e modele diferentes tipos de relatórios. Esta opção será aprofundada na seguinte seção (*Report Designer*)

Manda a seleção para o recurso de Etiquetas. As consultas criadas são a base para a criação de etiquetas. Você pode modelar diferentes tipos de etiquetas usando *Label Designer*.

Guarda a seleção numa Tabela. Existem vários tipos de formatos que você pode usar, exemplo Excel, texto, DBF, etc. O resultado das consultas poderá ser exportado ao clicar o botão exportar, definindo primeiramente o tipo do arquivo e o nome.

Através desta rotina você poderá enviar o resultado da sua pesquisa para o Excel e criar sofisticados gráficos usando as opções desta ferramenta.

Fig 235 Formatos de exportação da consulta.

Atenção ! Esta opção não está disponível nesta versão.

Criando e personalizando Relatórios – *Report Designer*

O relatório é um meio simples e fácil de apresentar informações da Base de Dados em documentos impressos. Nesta seção mostramos como elaborar e emitir relatórios utilizando o *Report Designer*.

Os relatórios têm dois componentes básicos: a fonte de dados e o layout. A fonte de dados, como foi vista na seção anterior, normalmente é uma ou mais tabelas, podendo ser filtradas, ordenadas e selecionadas. O layout – que descreveremos nesta seção - é a forma em que os elementos que compõe o relatório são dispostos na página.

As bandas de um relatório

Um relatório se constitui pelo menos em três bandas: *Encabeçamento*, *Detalhes* e *Rodapé*, que aparecem como bandas *default* em um relatório vazio. Você pode definir, além disso, um Título e criar várias bandas aninhadas, definindo grupos um dentro de outro.

A banda Detalhes é onde, tipicamente, são incluídos, em uma linha ou mais, os campos para cada registro da tabela que está sendo impressa.

Depois de definir os dados sega os seguintes procedimentos:

- Clicar no botão executar consulta.
- Clicar no botão Relatório (indicado por um ícone impressora).
- Aparecerá a janela de Layout. Veja na Fig 236.

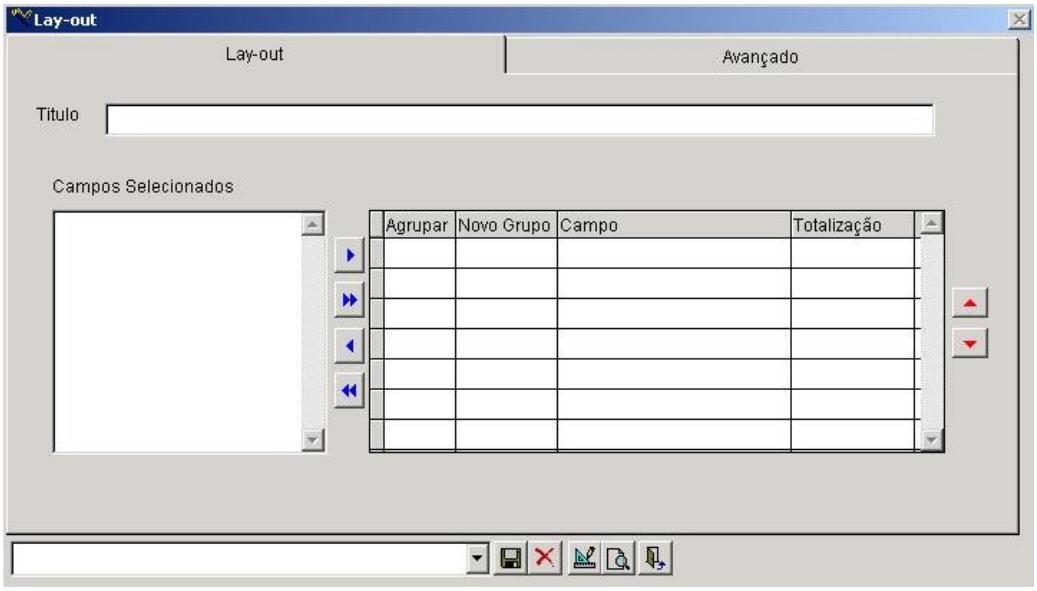


Fig 236 Janela Layout do Criador de Relatórios

Título:
Você usa a opção **Título** na janela Layout para criar a banda Título que encabeça o relatório.

Agrupamento:
Você usa as opções **Agrupar/NovoGrupo/Totalização** para agrupar em bandas os registros a serem impressos no relatório.

Aqui você cria ou modifica os grupos aninhados que separam os dados do relatório em bandas. Cada banda é constituída de um cabeçalho e de um rodapé, e ela é funcionalmente definida por um campo ou expressão que muda de valor quando relatório está sendo impresso

Os campos ou expressões selecionados para agrupamento definiram as quebras, cuja mudança de valor, durante a impressão do relatório, determina o início de uma nova banda. A ordem dos grupos nessa lista é muito importante, os grupos devem ir de pai para filho.

Por exemplo, num relatório sobre faturamento, você tem as tabelas relacionadas Clientes, Faturas e Itens da Fatura. A primeira banda se repete toda vez que um novo Cliente começa ser impresso; a Segunda, quando uma nova Fatura do Cliente (ordem) começa ser impressa; e os itens da Fatura são os Detalhes sendo impressos.

Visualizando e Imprimindo

Clicando sobre este botão com o mouse, você - em qualquer momento- pode visualizar como está ficando o relatório antes da impressão. Aparecerá a visualização acompanhada com os seguintes botões de opções padrão Windows:

Fig 237 Botões de Visualização e Impressão.

Personalizando o Relatório

Clicando neste botão, você personalizará o relatório incluindo rótulos, logos, ajustando a posição, etc. Esta opção chamara uma tela com as bandas definidas na opção do agrupamento. Veja na Fig 238 as três bandas *default* de um relatório vazio. Aparece também uma Caixa de

Ferramentas e uma janela de Propriedades, muito similar a descrita quando desenhamos os formulários das entidades (veja no Capítulo V).

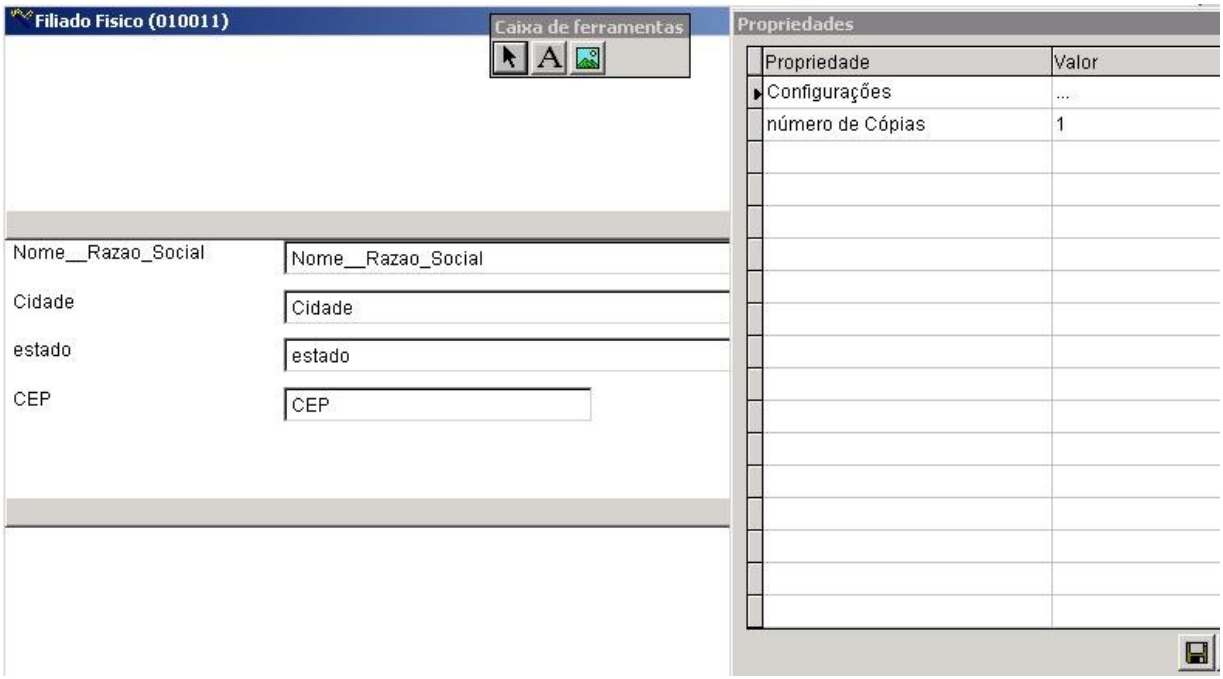


Fig 238 Personalizando Relatório.

A caixa de ferramentas permitirá inserir ou desenhar rótulos e imagens nos relatórios. Para adicionar:

- Selecione primeiro o objeto desejado na caixa de ferramentas fazendo um clique no ícone correspondente
- Leve o ponteiro do mouse para o local desejado dentro da banda específica do relatório.
- Dê um clique no mouse e o objeto será desenhado no lugar indicado.

Exemplo: Vamos inserir um logo no cabeçalho do relatório. Para incluir uma imagem basta:

- clicar no objeto imagem e selecionar a banda no relatório. Veja na seguinte figura.

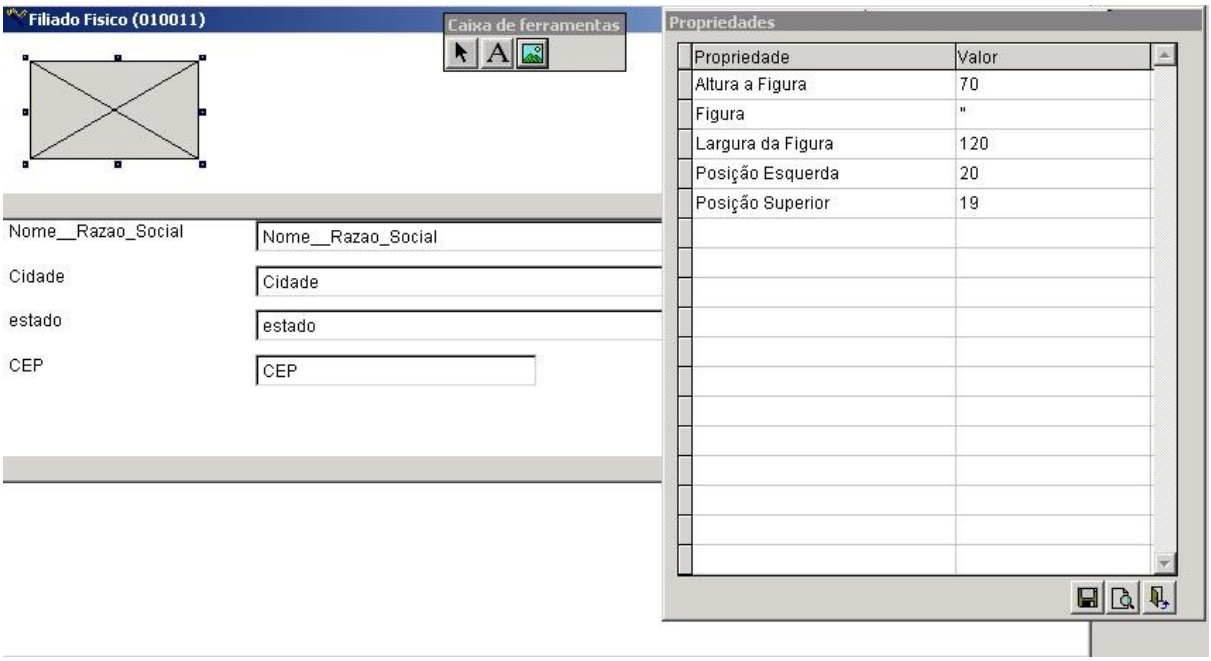


Fig 239 Inserindo uma imagem

- Clicar na propriedade Figura para definir o arquivo que contém a imagem.

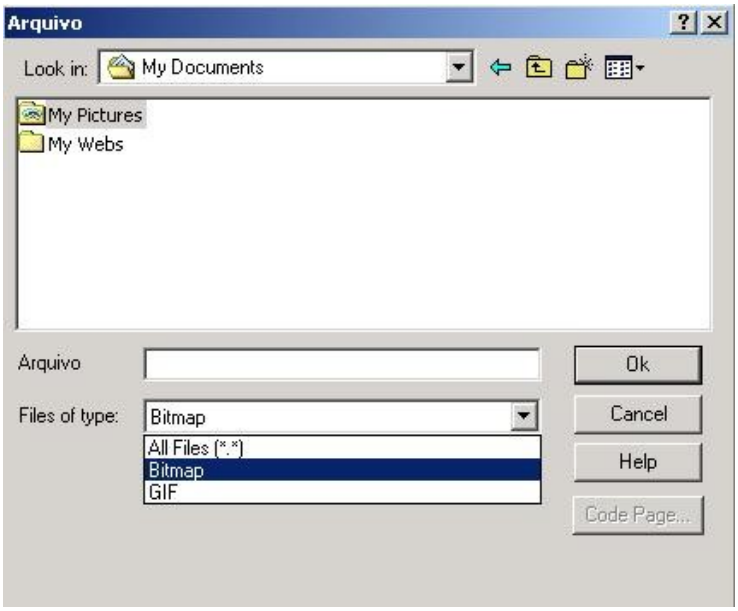


Fig 240 Selecionando o arquivo com a imagem.

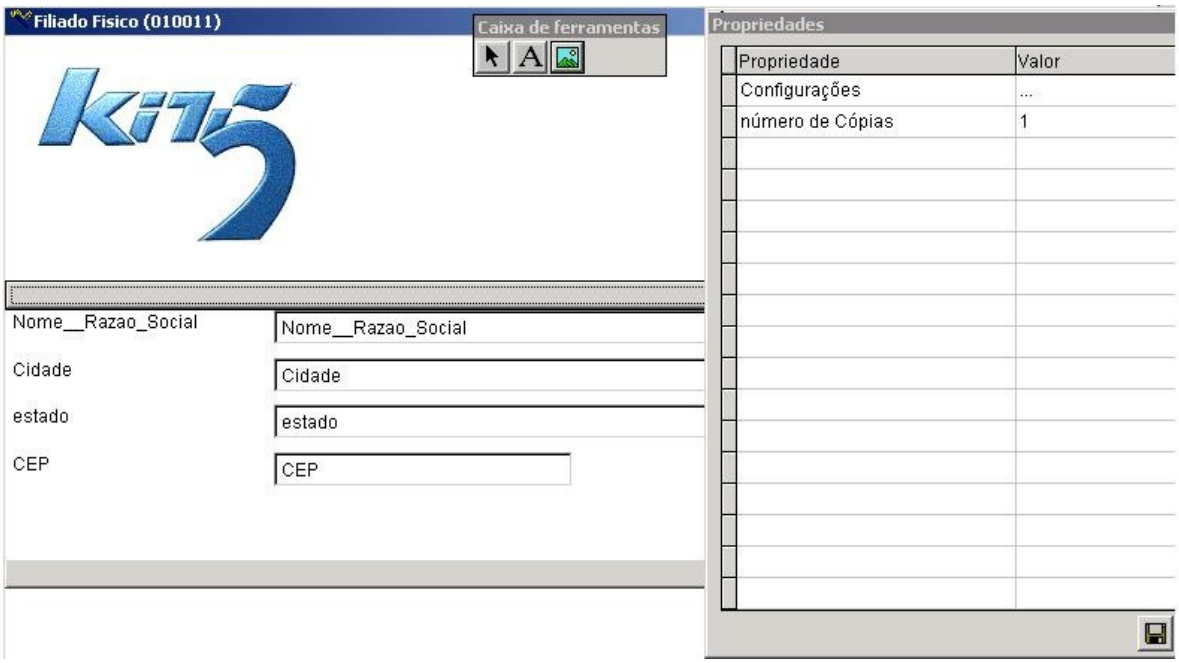


Fig 241 Imagem inserida no cabeçalho.

Fig 242 Relatório Personalizado

Você pode também, ajustar visualmente a altura de qualquer banda, clique no layout em cima da seta prefixado o nome da banda, e arraste verticalmente.

Criando e personalizando Etiquetas – *Label Designer*

A Etiqueta é um relatório multicolunas que coincide com algum papel de etiquetas particular. No Visual Kit 5, você pode criar etiquetas de uma maneira simples usando diferentes formatos ou configurações de etiquetas.

Depois de selecionada a origem de dados -explicada acima- você poderá acessar o *Label Designer* clicando no botão com o ícone de envelope.

Na seguinte figura é mostrada a janela da interfase do criador de etiquetas. Na parte superior da janela é mostrada uma lista com as diferentes configurações de etiquetas. As etiquetas são cadastradas para o dado fabricante selecionado na entrada de dados Fabricante. Selecione a etiqueta clicando encima de sua configuração. Automaticamente será mostrado a forma na área inferior à direita.

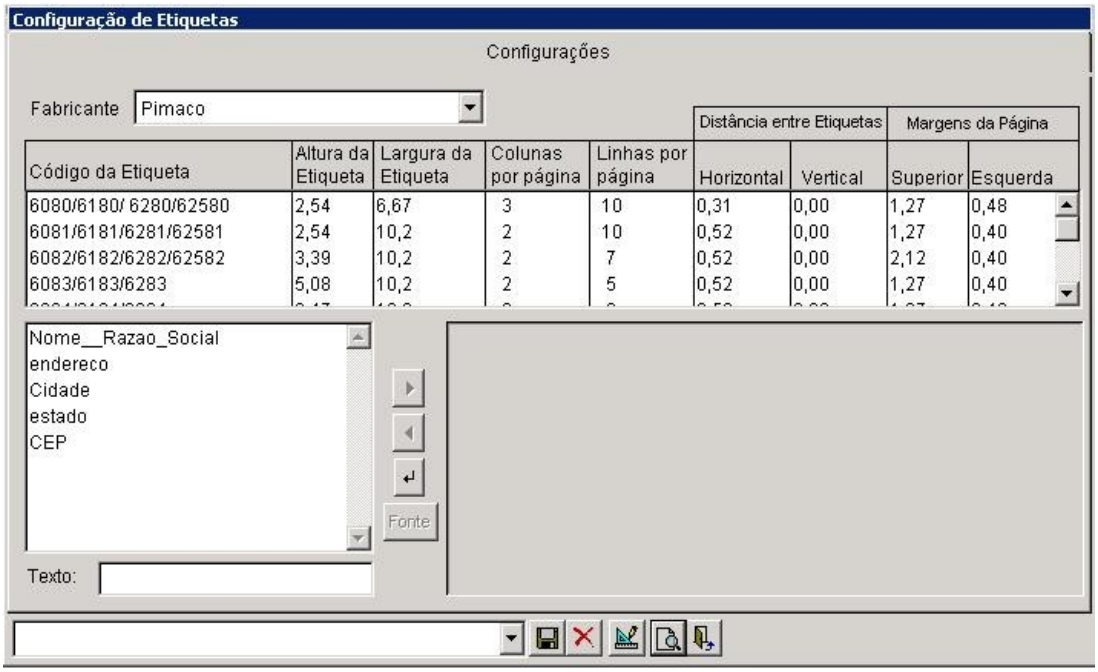


Fig 243 O *Label Designer*

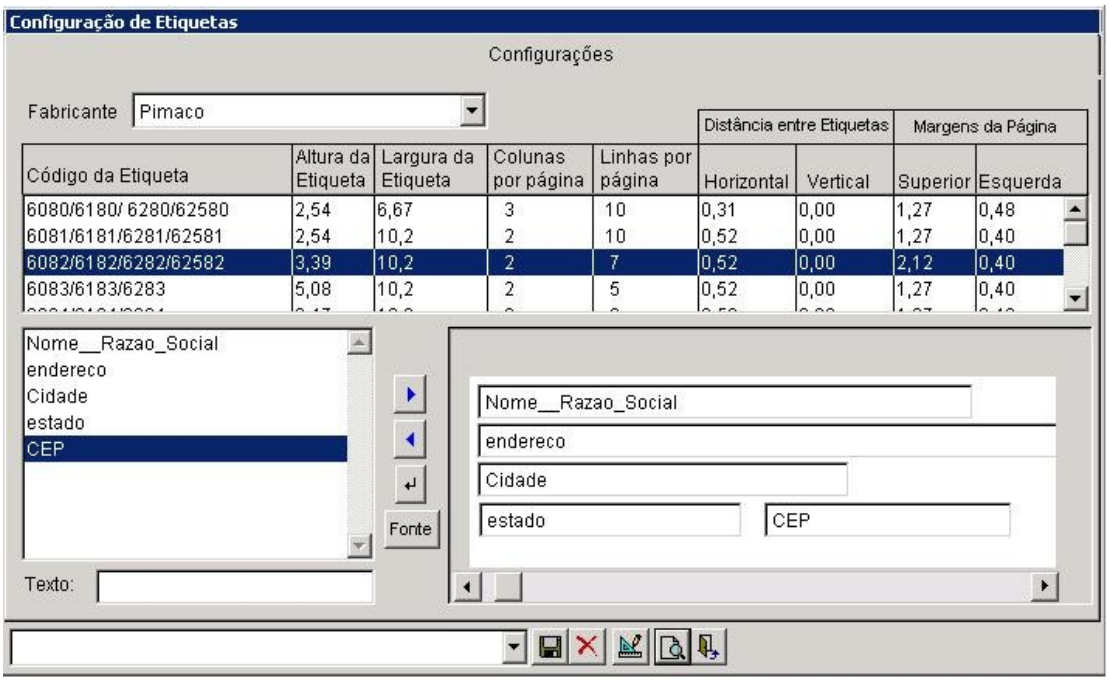


Fig 244 Personalizando a Etiqueta

Fig 245 Resultado final da geração das etiquetas.

Na lista à esquerda são mostrados os campos selecionados no passo Origem de Dados. Você poderá inserir estes campos na ordem desejada, assim como rótulos usando os botões.

Criando Códigos de Barras

Uma opção muito usada em varias aplicações é a geração dos códigos de barras, principalmente em relatórios ou etiquetas. Criar códigos de barras nos sistemas gerados pelo Visual Kit 5 é muito fácil. Vejamos os procedimentos.

Instalar Fonte

O primeiro passo a realizar é a inclusão de uma fonte tipo código de barra – caso não exista - no computador onde o sistema gerado vai rodar. Este passo é realizado uma única vez. Para isso:

- Dê um clique no botão **Iniciar** (*Start*) na barra de tarefas (*Task Bar*) do seu Windows (ou pressione <CTRL>+<ESC>).
- No menu apresentado, selecione **Configurações** (*Settings*).
- Na janela aberta clique no **Painel de Controle** (*Control Panel*)

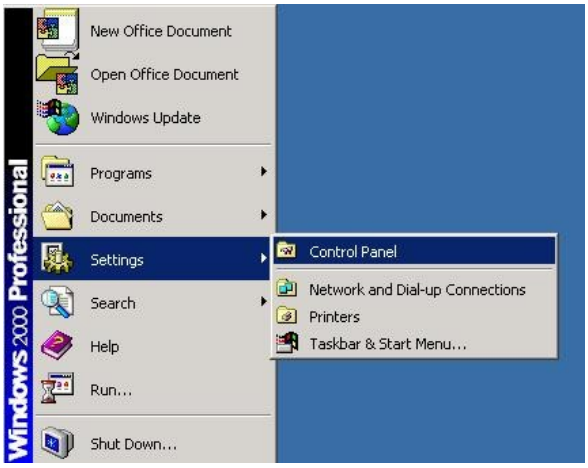


Fig 246 Selecionando o Painel de Controle

- Será aberta a janela da Fig 247 com todas as pastas do **Painel de Controle**.
- Selecione a pasta **Fontes** (*Fonts*), indicada na figura com o circulo em vermelho.
- Automaticamente será aberta uma janela com todas as fontes usadas no computador (Veja na Fig 248).
- Clique em **Arquivo** (*File*) e selecione **Instalar Nova Fonte...** (Install New Font...).

Fig 247 Pastas do Controle de Painel

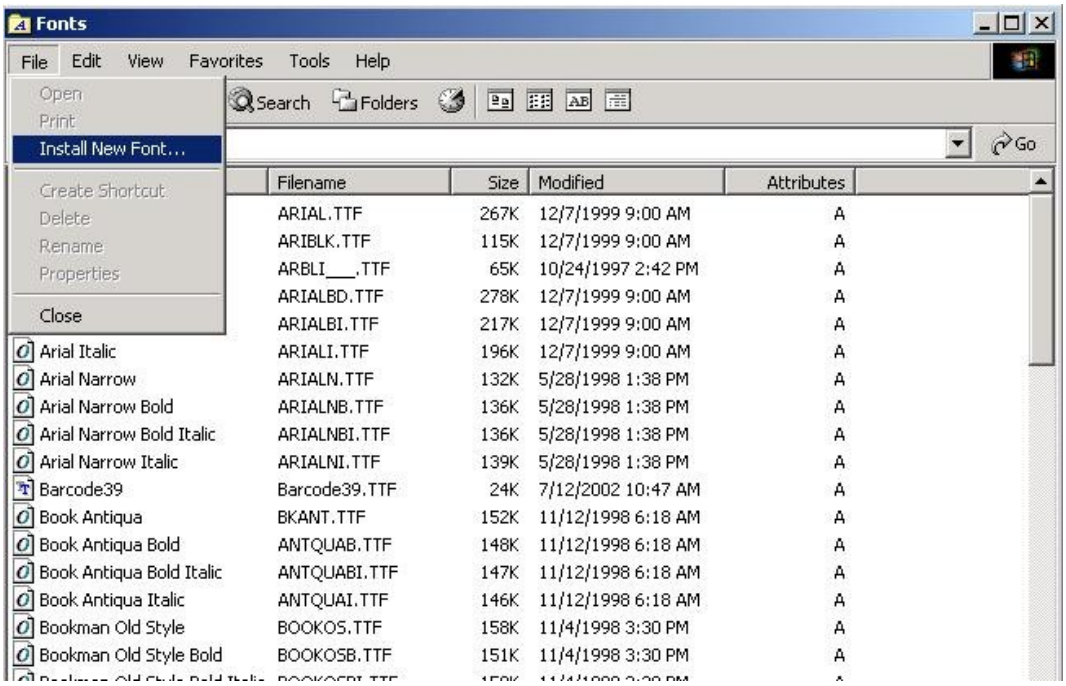


Fig 248 Inserindo uma nova fonte

- Indique o caminho da nova fonte **Barcode39** que se encontra na pasta do sistema gerado, (ex. Kit_clientes), dentro do "**parameters"**, na pasta "**font"**. Ou seja c:\Kt_clientes\parameters\font
- Selecione a fonte **Barcode39** e clique no botão **OK**.
- Automaticamente será instalada a fonte. Veja na Fig 250 a fonte já instalada.

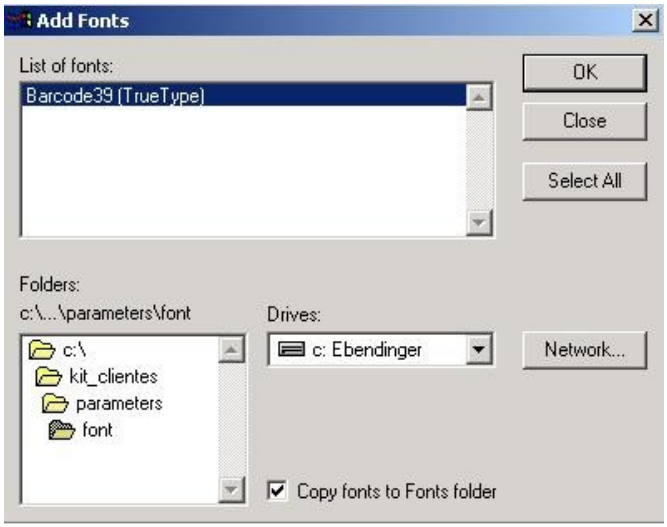


Fig 249 Selecionando a nova fonte

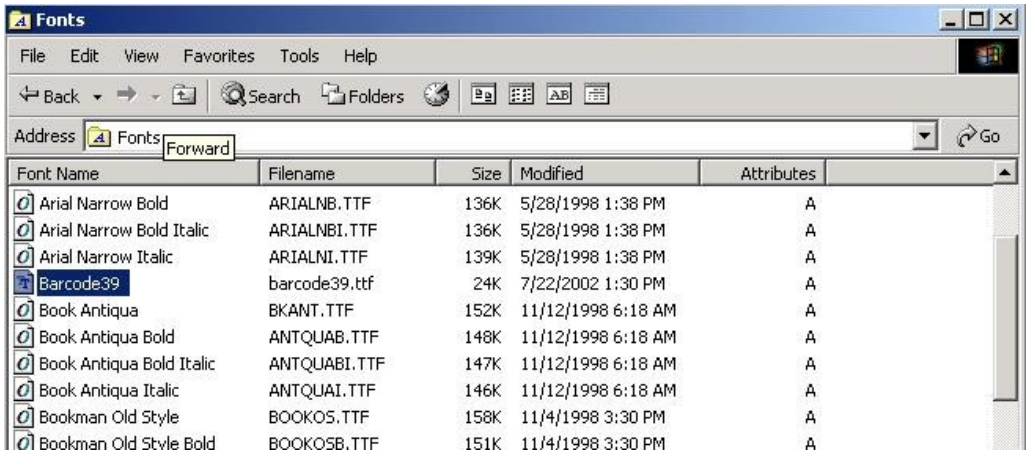


Fig 250 Barcode instalado.

Criando o Código de Barra

Vamos a mostrar o uso desta opção, gerando etiquetas com código de barras. Veja os procedimentos:

- Selecione a operação **Consulta, Relatório e Etiquetas Avançadas** como vimos na seção anterior.
- Escolha a entidade ou arquivo com o campo que deseje converter em código de barras.
- Clicando no botão abra o **Construtor de Expressões** mostrado na Fig 252 para criar uma nova expressão que converterá o campo em código de barras.

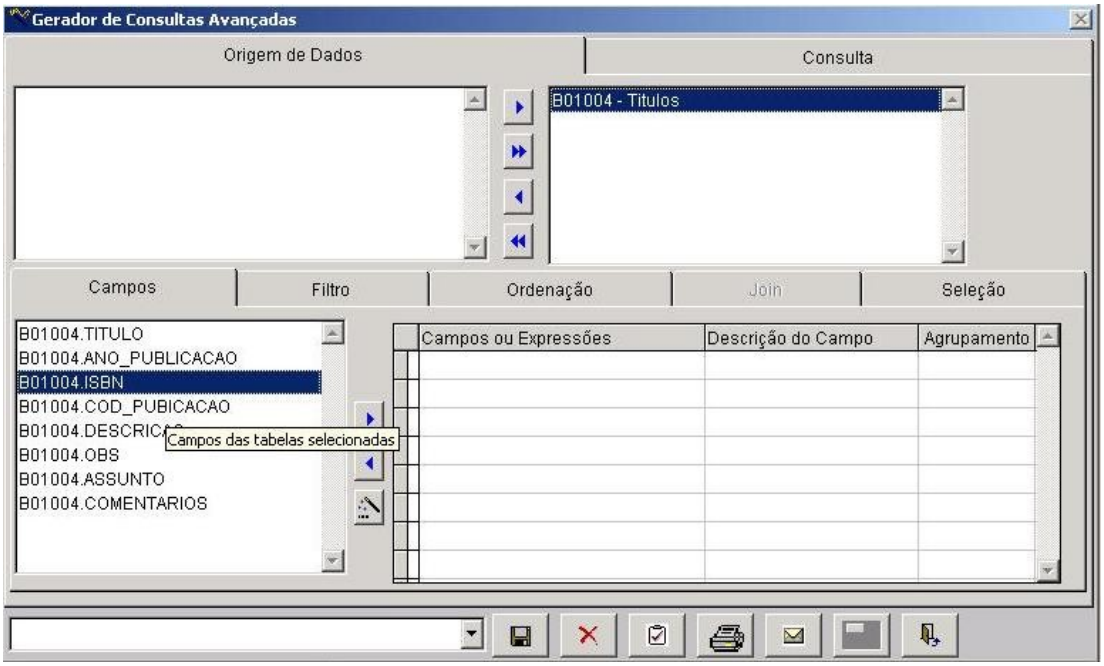


Fig 251 Escolhendo a Origem de Dados

- Dentro do **Construtor de Expressões**, clique encima do campo que você deseja converter para código de barras (ex ISBN) e selecione a função preestabelecida **Código de Barra campo caractere** dando um duplo-clique.
- Automaticamente será preenchida a expressão mostrada na Fig 253 que realizará a conversão. Digite um nome para esta expressão (ex. *Barras*) e clique no botão **OK**.

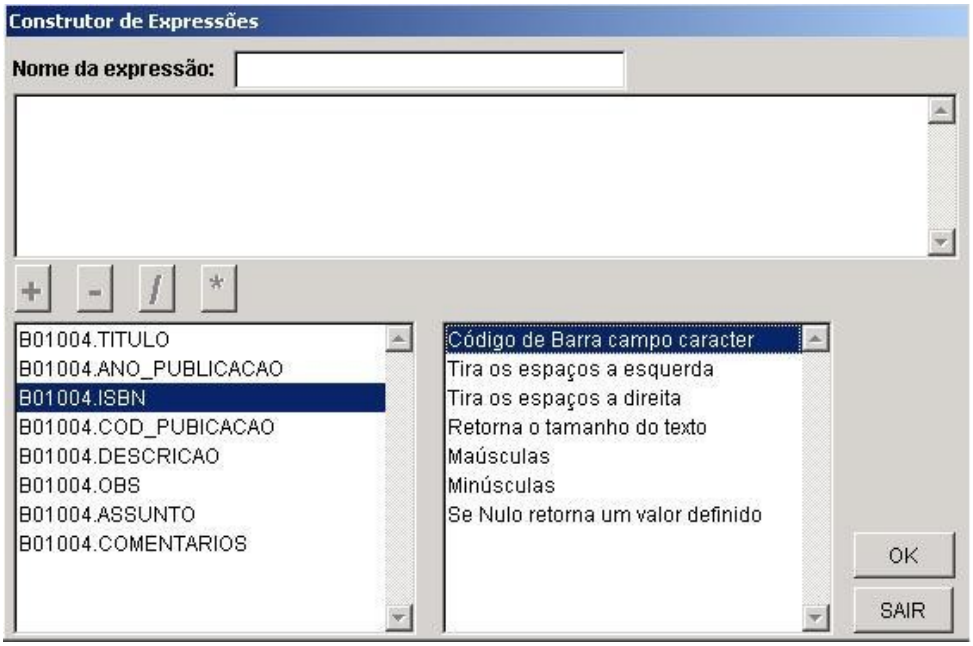


Fig 252 Construtor de expressões

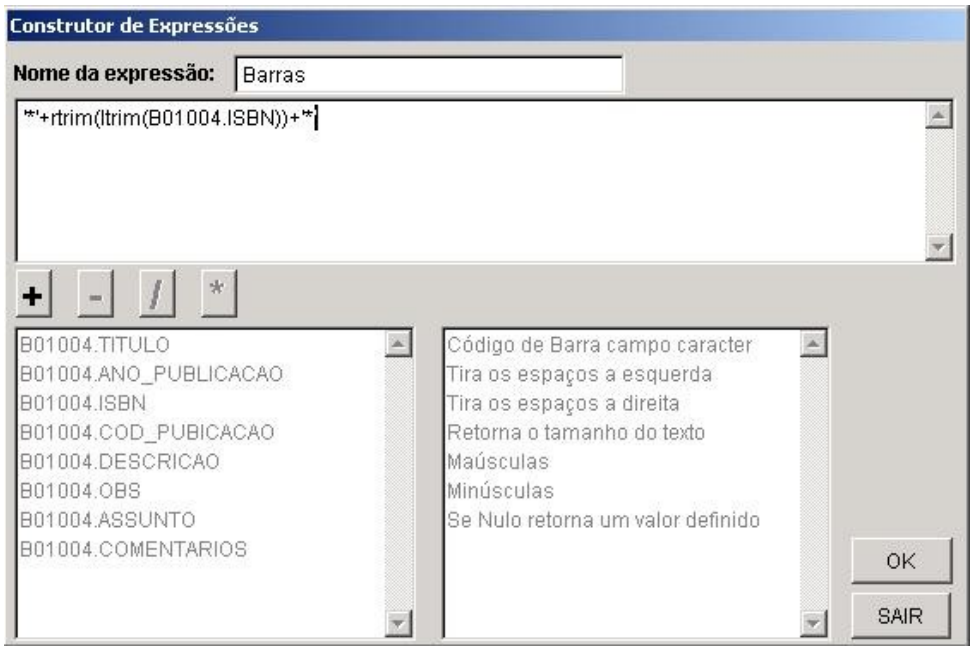


Fig 253 Expressão para código de barras.

- Veja nesta figura como ficou a origem de dados depois da criação da expressão. Você pode inserir ou não outros campos (ex. TITULO).
- Clique no botão para executar Consulta.

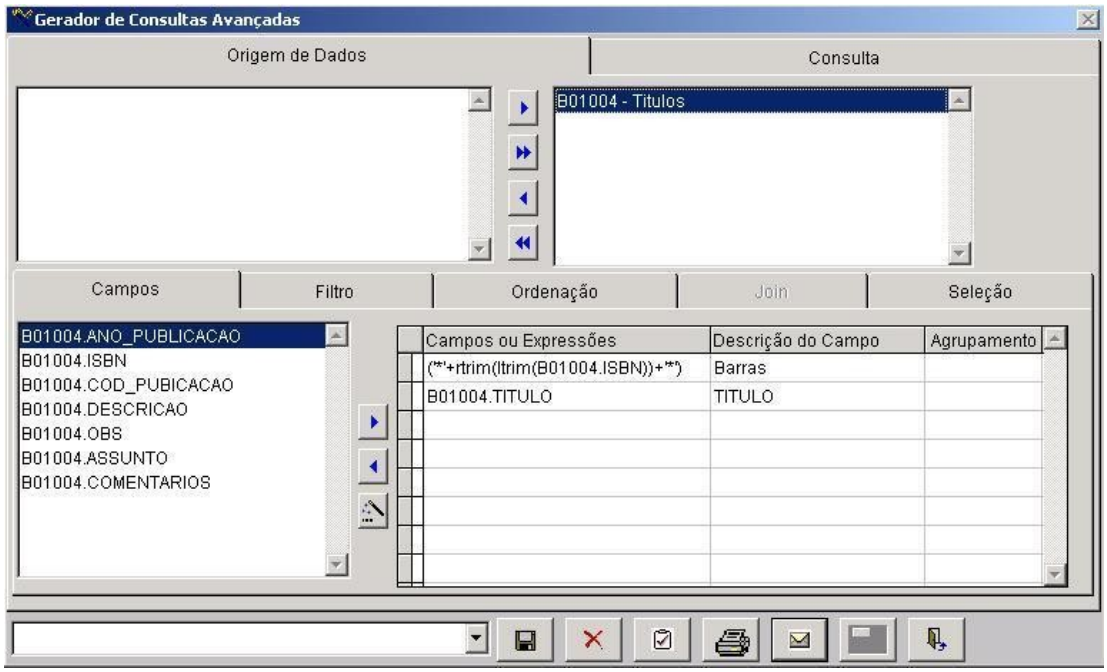


Fig 254 Origem de dados com expressão de conversão do código de barras.

Atenção! Antes de executar um Relatório ou uma Etiqueta é preciso primeiramente executar a Consulta.

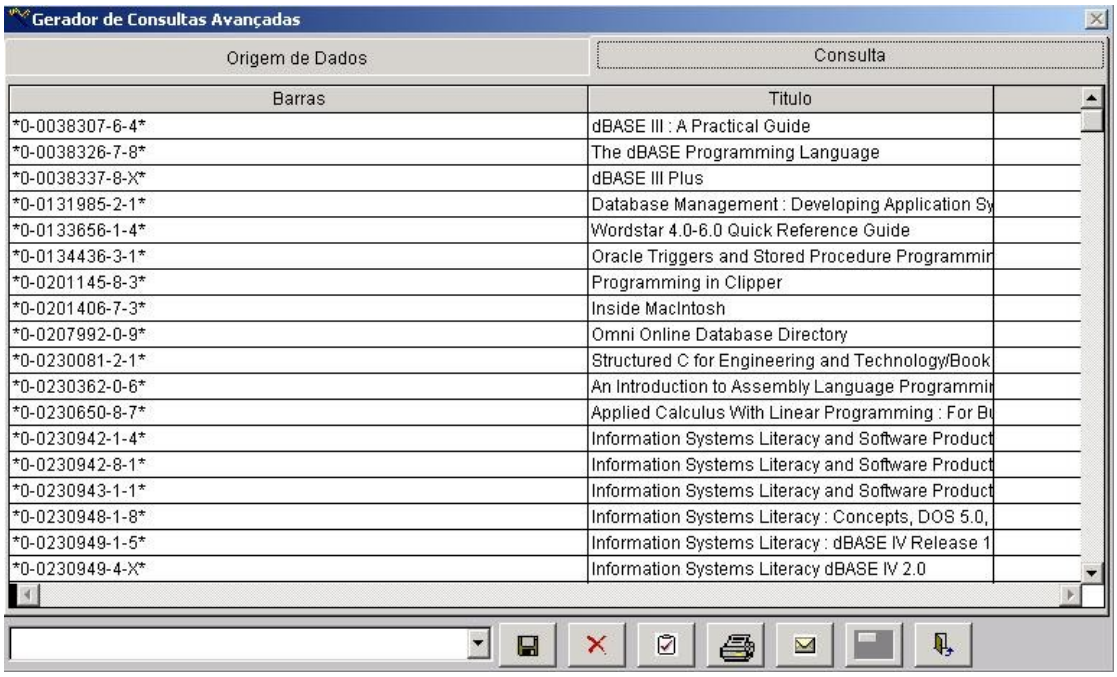


Fig 255 Consulta executada.

- Clique no botão de criar etiquetas como descrito na seção anterior.
- Selecione os campos e personalize a etiqueta
- Clique encima do campo com o código de barra (ex. *Barras*) e clique no botão **Fonte**.
- Aparecerá a janela da Fig 257 onde selecionaremos a fonte cadastrada no passo anterior, ou uma outra fonte de tipo de código de barras.
- Por último clique no botão visualizar para ver o resultado das etiquetas com código de barras. Veja na Fig 258.

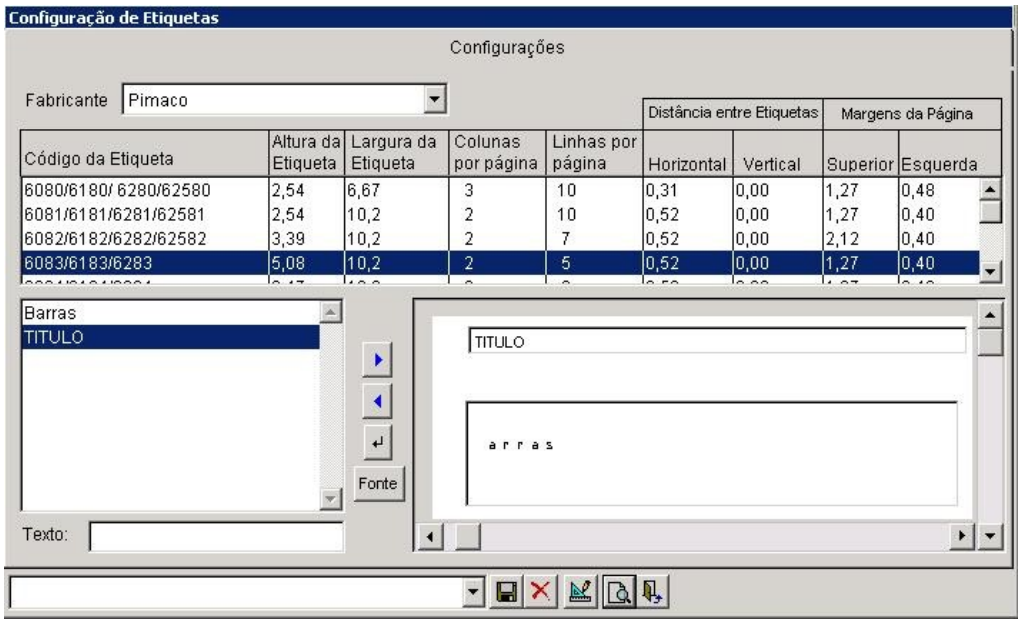


Fig 256 Personalizando o formulário

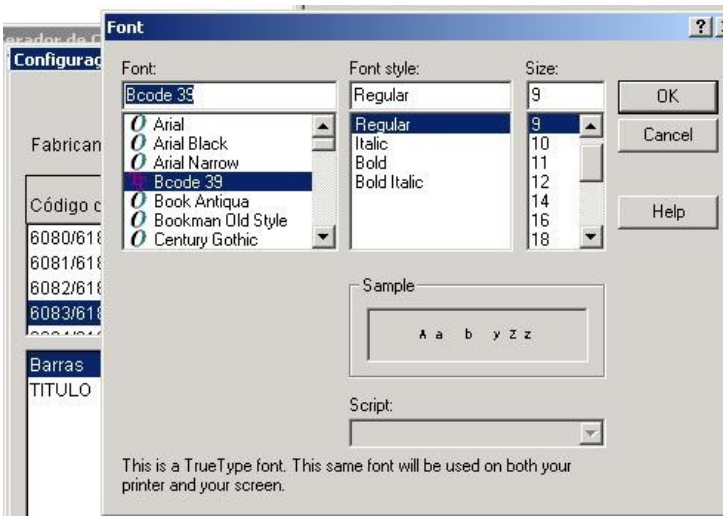


Fig 257 Selecionando a fonte.

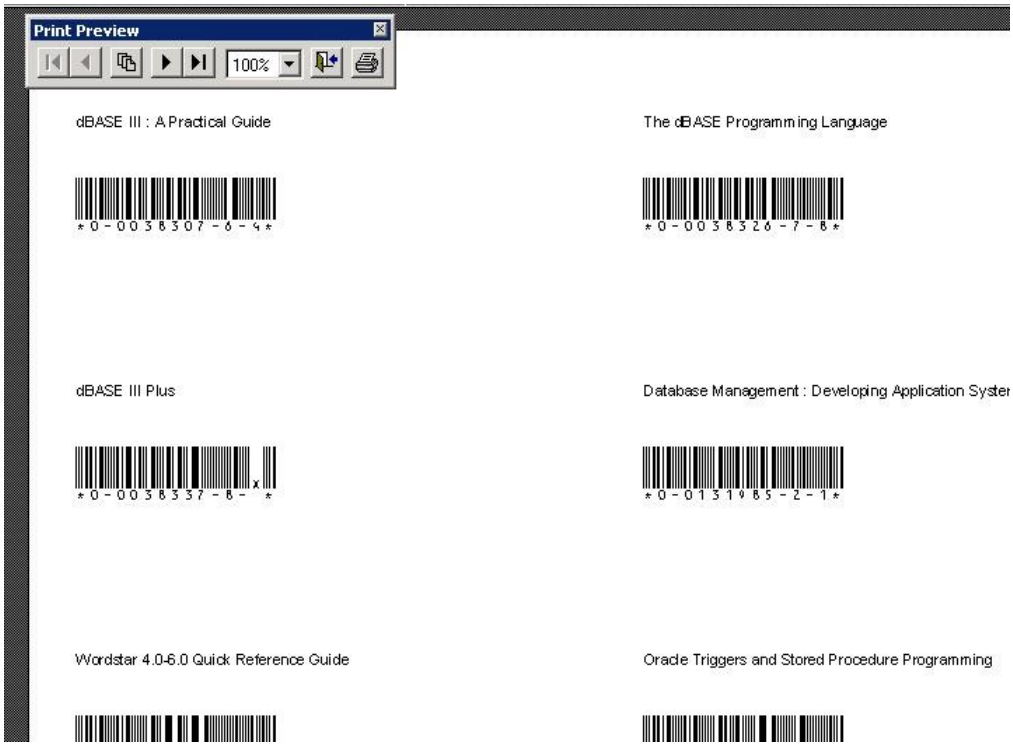


Fig 258 Resultado final

OLAP- Processamento Analítico *Online*

O recurso ***O**n-**L**ine **A**nalytic **P**rocessing* ou processamento analítico on-line proporciona as condições de análise de dados para responder perguntas com enfoque mais estratégico e de maneira simples e rápida.

O que é OLAP?

O OLAP é uma maneira especial de colocar ou modelar os dados das tabelas do sistema. De um ponto de vista prático, usando o *OLAP,* o próprio usuário realiza consultas *interativas* aos dados, seguindo um caminho de análise através de múltiplos passos, como, por exemplo: aprofundar-se sucessivamente por níveis mais baixos de detalhe de um quesito de informação específico.

No *OLAP* os dados são processados em um formato chamado de *Cubos-OLAP* ou simplesmente de *Cubos*. Este modelo permite realizar análises multidimensionais (cruzamento de varias informações) de uma maneira rápida. Vejamos na Fig 259 um modelo esquemático de um *Cubo-OLAP*.

Neste exemplo, o cubo está formado por três dimensões: Geográfica, Produtos e Tempo. (o número de dimensões pode maior ou menor – colocamos três para dar uma explicação melhor do conceito). Cada célula do cubo guarda uma informação que podem ser totalizadas em níveis superiores ou desmembradas em níveis inferiores. Exemplo os niveis Estado->Ciudade->Bairro.

Navegar e manipular as células do cubo pode ser feita através de uma tabela dinâmica (*Pivot Table*), onde as linhas e colunas são conformadas e mudadas de acordo as necessidades do usuário com simples cliques do mouse.

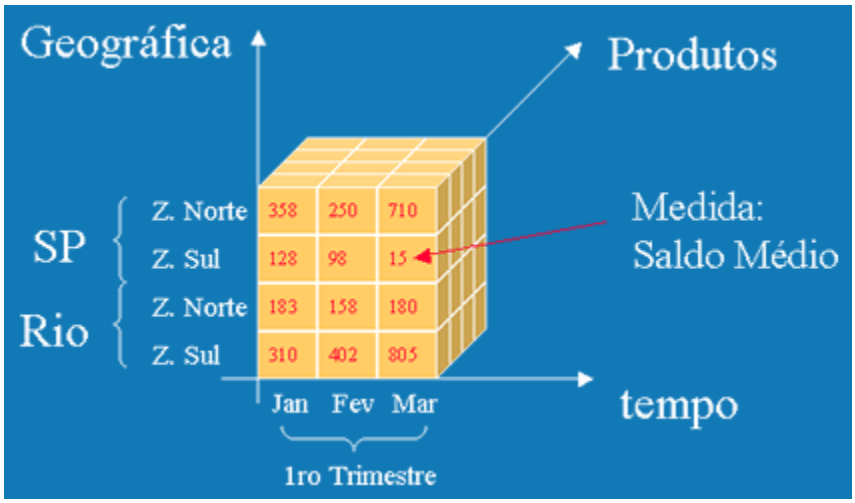


Fig 259 Modelo esquemático dos Cubos OLAP

Procedimentos para usar o OLAP

Na janela OLAP você encontrará três páginas: Origem de Dados, Criação de Cubo e OLAP. Vejamos a seguir a finalidade de cada uma.

Origem de Dados

Na janela **Origem de Dados** você identifica primeiramente as entidades (tabelas), que serão usadas para processar ou criar o cubo. Os campos das entidades são logo carregados na lista **Campos,** para a seleção. Selecionados os campos que serão usados, você pode clicar na próxima página: **Criação do Cubo**.

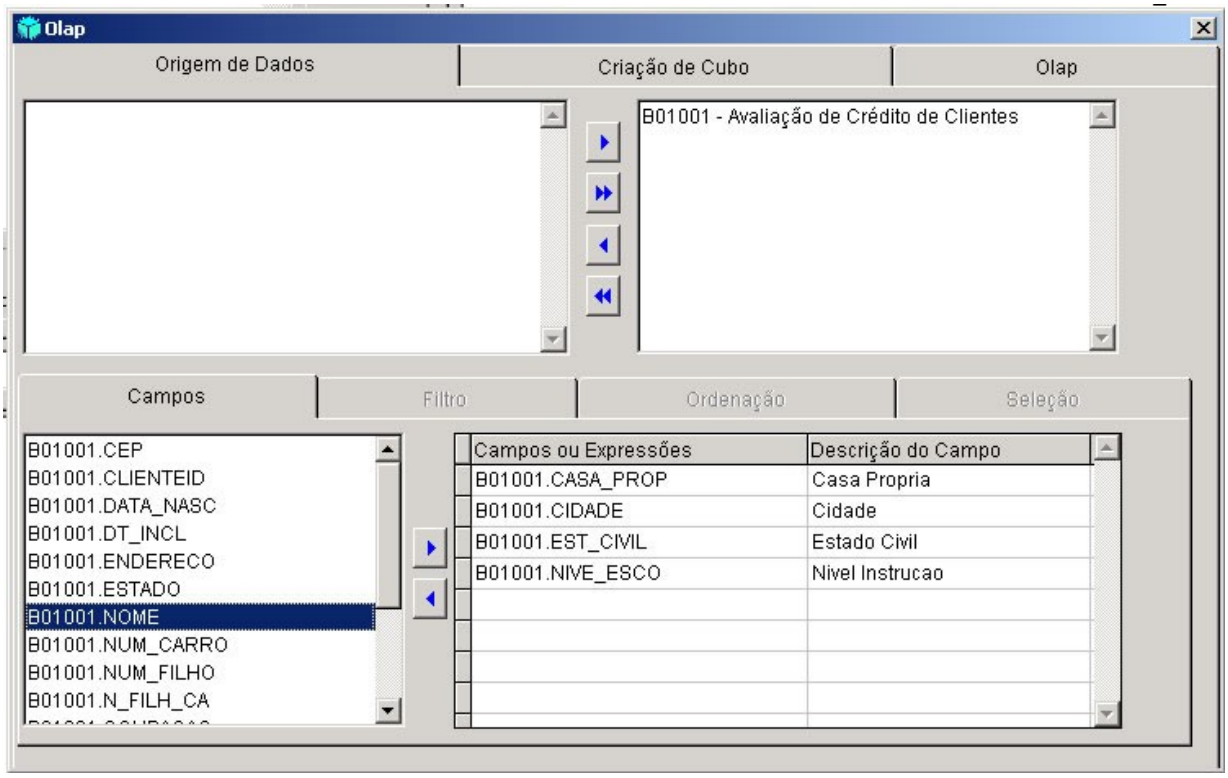


Fig 260 Janela de seleção dos Dados para gerar os cubos OLAP

Criando o Cubo

Na página **Criação de Cubo** você escolherá quais campos serão as dimensões (campos para o cruzamento) e quais serão totalizações. Totalização é o parâmetro que será analisado e as dimensões, as variáveis para fazer os cruzamentos de dados na pesquisa OLAP. Por exemplo:

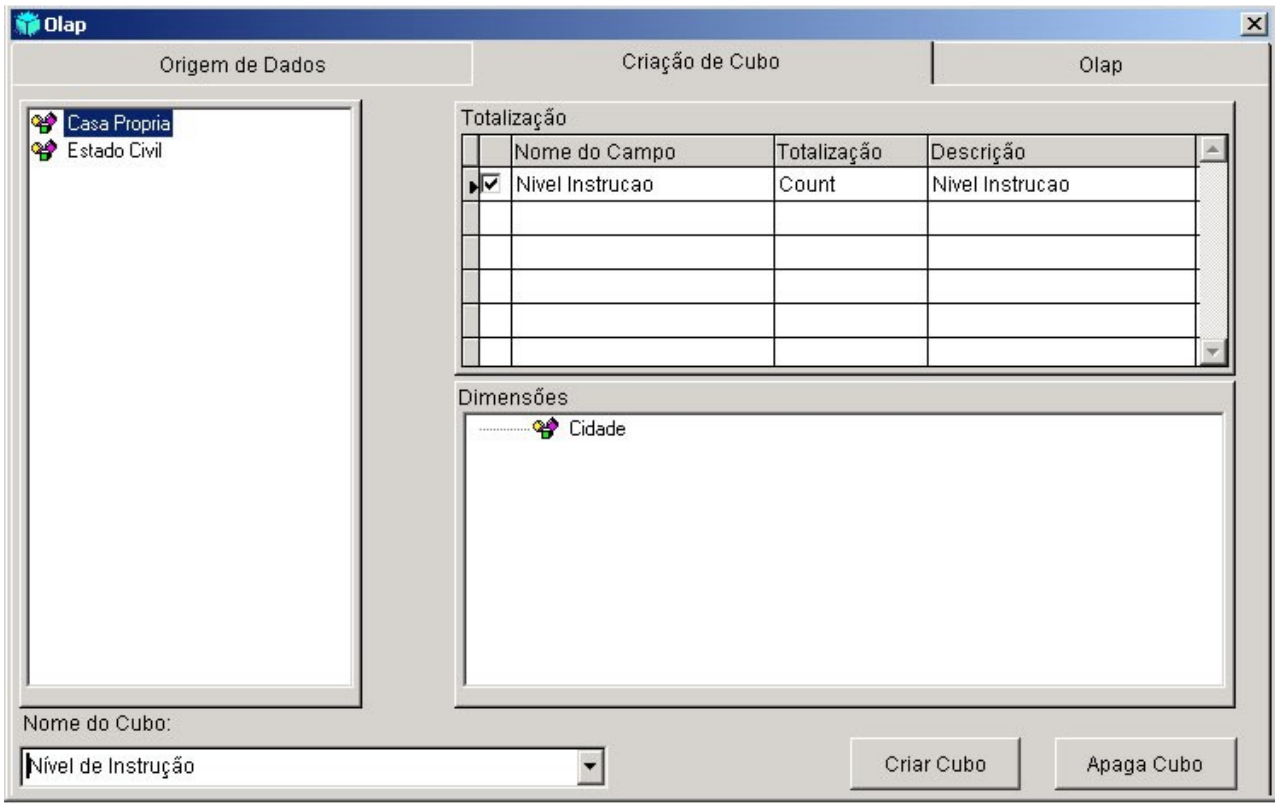


Fig 261 Escolhendo as Totalizações e as Dimensões.

Você poderia indicar - selecionando e arrastando com o mouse- uma totalização igual a *quantidade de clientes (*Count*)*, e as dimensões S*exo*, *Profissão*, *Casa Própria*, *Carro* e *Cartão*.

Na entrada de dados **Nome do Cubo**, coloque um nome para o cubo que vai ser criado.

Clique no botão **Criar Cubo** e o cubo será processado e gerado. Você pode apagar cubos anteriores que não estejam sendo usados, selecionando-o e clicando no botão **Apaga Cubo**.

Nota

Você pode criar dimensões com níveis hierárquicos - exemplo Estado->Cidade->Bairro – mantendo sempre a ordem da hierarquia. Para isso, basta com o mouse arrastar o campo de nível inferior para embaixo do campo de nível superior de forma que crie uma estrutura em árvore.

Em dependência da quantidade de dados e do número de dimensões e totalizações o processamento do cubo pode demorar. Será apresentada uma mensagem indicando que o cubo está sendo processado.

Bom, depois do processamento, chegou a hora de navegar pelo cubo e fazer as pesquisas desejadas. Clique para na página de **Olap**.

Navegando pelo Cubo

Na página OLAP (Fig 262) você poderá realizar as pesquisas usando como interface uma tabela dinâmica. Os procedimentos são os seguintes.

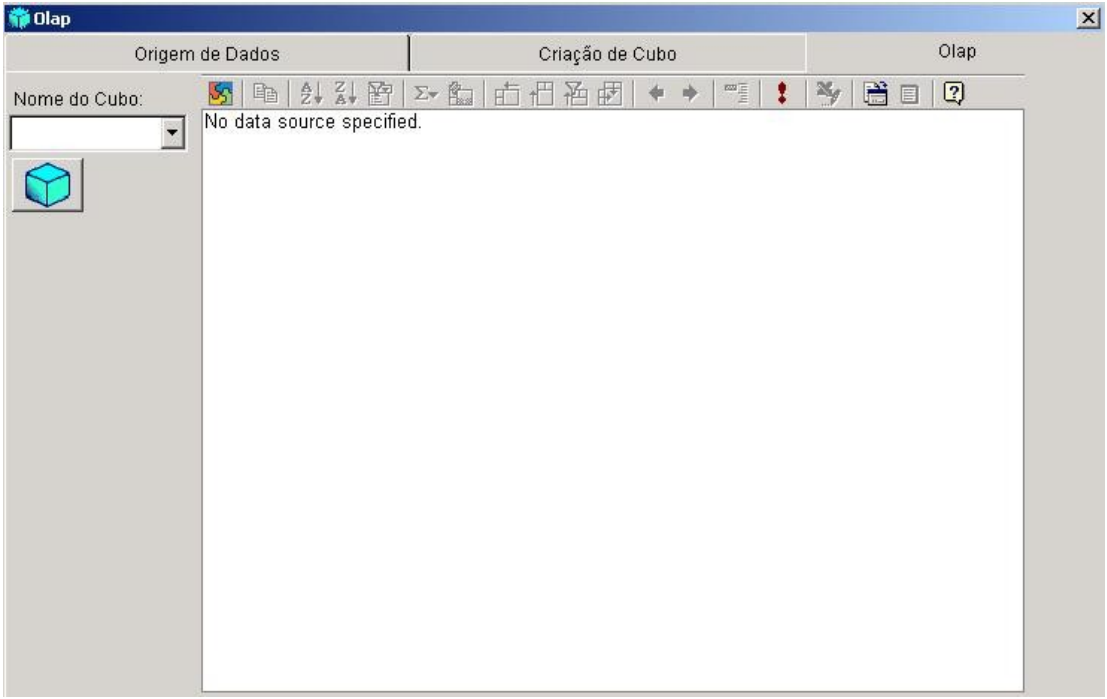


Fig 262 Pagina de analise e navegação pelo cubo

- Na entrada de dados **Nome do Cubo** selecione o nome do cubo processado.
- Para carregar o cubo, clique no botão com a imagem de um cubo azul.
- Será apresentada uma tabela dinâmica. Esta tabela possui um menu com varias opções.
- Selecione a opção indiada na Fig 263, para aparecer aberta a lista das dimensões e totalizações do cubo.
- Selecionando e arrastando com o mouse, coloque os items da lista aberta na tabela dinâmica da forma desejada. Veja na Fig 264

Você pode utilizar as ferramentas apresentadas na tabela dinâmica como por exemplo carregar o cubo no programa Excel para a geração de gráficos. Para maiores informações pressione o botão **Help** da janela.

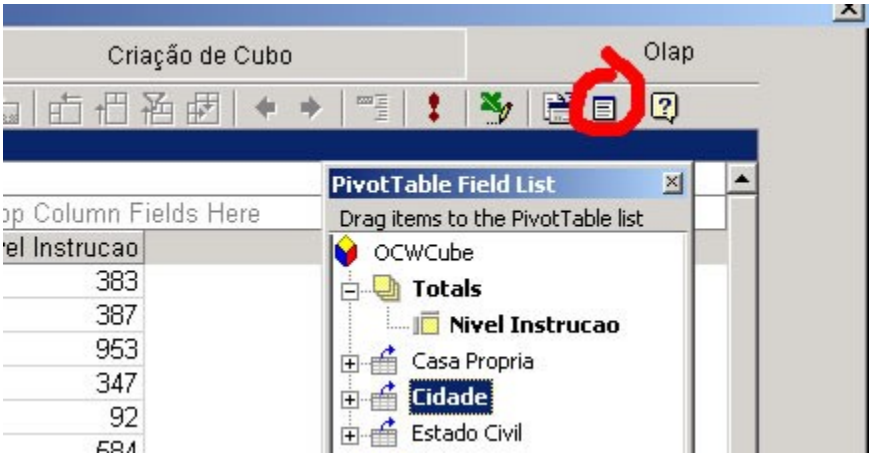


Fig 263 Selecionando a lista das dimensões.

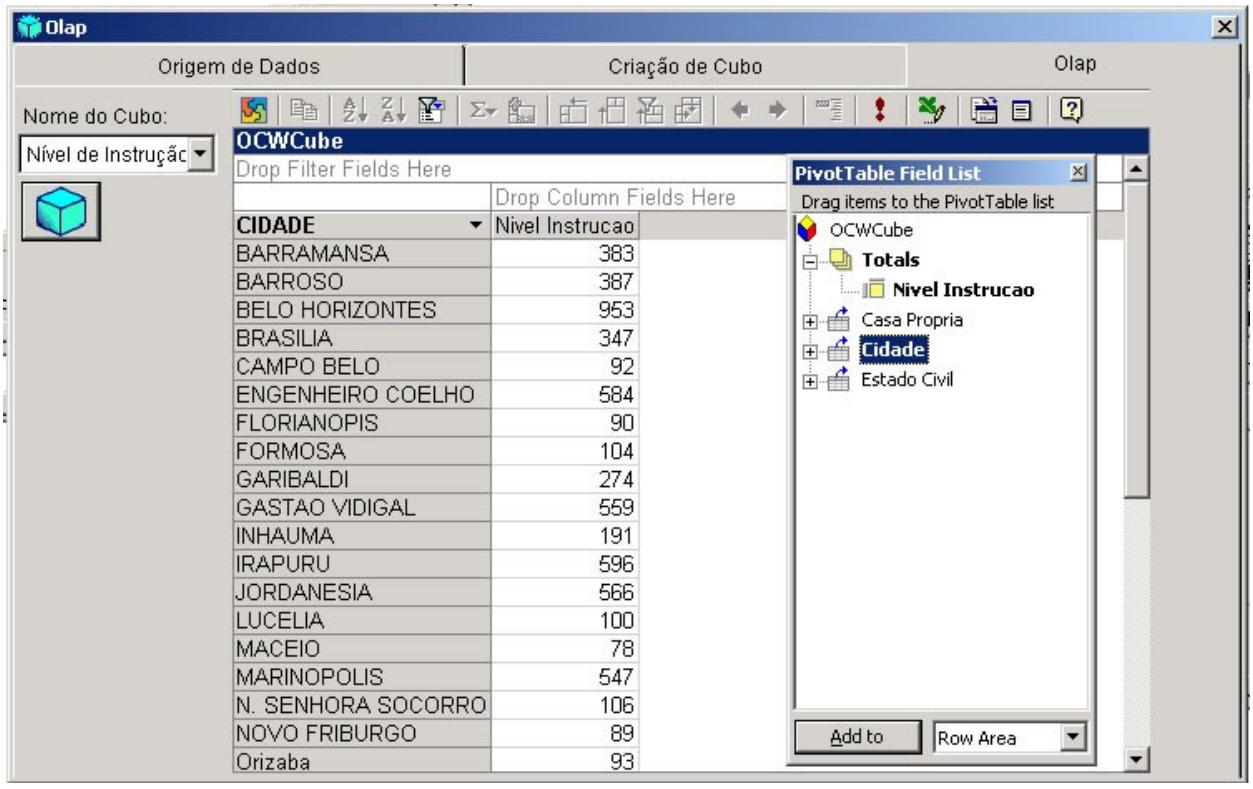


Fig 264 Navegando pelo Cubo

Principais Definições

Dentro do navegador do cubo existe uma opção de ajuda (**Help**) com uma explicação bastante detalhada do manuseio da tabela dinâmica. Esta ajuda

pode aparecer em idioma inglês, por isso, colocamos aqui algumas definições importantes.

Drill Across: O *Drill Across* ocorre quando o usuário pula um nível intermediário dentro de uma mesma dimensão. Por exemplo: a dimensão tempo é composta por ano, semestre, trimestre, mês e dia. O usuário estará executando um *Drill Across* quando ele passar de ano direto para semestre ou mês.
Drill Down: O *Drill Down* ocorre quando o usuário aumenta o nível de detalhe da informação, diminuindo o grau de granularidade.
Drill Up: O *Drill Up* é o contrário do *Drill Down*, ele ocorre quando o usuário aumenta o grau de granularidade, diminuindo o nível de detalhamento da informação.
Slice And Dice: O *Slice and Dice* é uma das principais características de uma ferramenta *OLAP*. Ele serve para modificar a posição de uma informação, alterar linhas por colunas de maneira a facilitar a compreensão dos usuários e girar o cubo sempre que tiver necessidade.
Alertas: Os *Alertas* são utilizados para indicar situações de destaque em elementos dos relatórios, baseados em condições envolvendo objetos e variáveis. Servem para indicar valores mediante condições, mas não para isolar dados pelas mesmas.
Ranking: A opção de ranking permite agrupar resultados por ordem de maiores / menores, baseado em objetos numéricos (*Measures*).Esta opção impacta somente uma tabela direcionada (relatório) não afetando a pesquisa (*Query*).
Filtros: Os dados selecionados por uma *Query* podem ser submetidos a condições para a leitura na fonte de dados. Os dados já recuperados pelo Usuário podem ser novamente “filtrados” para facilitar análises diretamente no documento.
Sorts: -Os *sorts* servem para ordenar uma informação. Esta ordenação pode ser customizada, crescente ou decrescente.

Data Mining – Inteligência Artificial

Data Mining ou *Exploração de Dados* é um ramo da informática que utiliza técnicas de estatísticas e inteligência artificial para construir modelos matemáticos.

Data Mining é o processo de extrair informação válida, previamente desconhecida e de máxima abrangência a partir de grandes bases de dados, usando-as para efetuar decisões cruciais. O *Data Mining* vai muito além da simples consulta a um banco de dados, no sentido de que permite

aos usuários explorar e inferir informação útil a partir dos dados, descobrindo relacionamentos escondidos no banco de dados. Pode ser considerada uma forma de descobrimento de conhecimento em bancos de dados.

Definição

Data Mining difere de técnicas estatísticas porque, ao invés de verificar padrões hipotéticos, utiliza os próprios dados para descobrir tais padrões. Por exemplo, supondo que o analista que queria identificar os fatores de risco, utilizasse uma ferramenta de *Data Mining*. A ferramenta poderia descobrir que mulheres adultas são clientes de risco, mas a ferramenta pode ir além e descobrir um padrão que o analista não pensou inicialmente. Por exemplo, que o tipo de estado civil e número de filhos em combinação com a idade da mulher também são determinantes do risco.

Onde utilizar o Data Mining.

O *Data Mining* pode ser utilizado com os seguintes objetivos:

-*Explanatório*: explicar algum evento ou medida observada. Ex: Por que a aberturas de novas contas subiram no Rio de Janeiro?;
-*Confirmatório*: confirmar uma hipótese. Ex: pode-se querer examinar os registros dos clientes para determinar se famílias de duas rendas tem mais probabilidade de adquirir um seguro de vida do que famílias de uma única renda;
-*Exploratório*: analisar os dados buscando relacionamento novos e não previstos. Ex: analisar registros históricos para determinar que fatores estão associados a clientes que devolvem um produto.

Entre as aplicações típicas do uso das tecnologias de *Data Mining* se destacam: Predição do lucro, gastos, etc; Aquisição de novos clientes; Retenção de clientes; Detecção de Fraudes; etc.

Recursos de Data Mining no Visual Kit 5.

O recurso de Data Mining implementado na versão Standard do Visual Kit 5 é um recurso muito usado na criação de modelo preditivos. Ele é composto de um único algoritmo chamado de *Microsoft DecisionTrees* (Arvores de Decisão). Este algoritmo determina as regras para predizer o conteúdo de um determinado campo – chamado também de campo de predição ou de saída do modelo- em dependência dos valores de outros campos - (chamados de campos de entrada do modelo).

Nota

Sai do marco deste livro a explicação do modelo de *Microsoft DecisionTrees*. Informações detalhadas deste algoritmo poderão ser encontradas na página da Microsoft.
Em próximas atualizações do produto serão incluídos novos algoritmos de Data Mining.

No recurso de Data Mining você encontrará três páginas: Origem de Dados, Criação do Modelo e Treina Modelo. De maneira geral você realizará:

- Passo 1: Você seleciona primeiramente uma tabela contendo os casos, de onde o algoritmo de Data Mining vai extrair as informações;
- Passo 2: Você define os campos de entrada do modelo e o campo que vai ser predito;
- Passo 3: A tabela de casos é dividida em duas partes: uma parte dos registros são utilizados para criar o modelo, ou seja achar as regras de predição; a outra parte será utilizada para testar as regras. Não teria poder de generalização testar o modelo com os mesmos dados que o modelo foi criado;

Na fase de teste, é calculado um valor de erro que, quanto menor seja este valor, melhor fica o modelo;

- Passo 4: Escolha do melhor modelo – na prática você criará vários modelos, e escolhe o que menor valor de erro possuía;

Selecionado o melhor modelo, você o embute dentro do formulário para predizer novos conteúdos do campo. Por exemplo:

Imagine que você tem um negócio que outorga aos clientes um cartão que o identifica como cliente VIP, Executivo ou Standard. Agora, suponha que você queira que quando seja preenchido o formulário de cadastro de um novo cliente, o sistema prediga automaticamente -segundo as características do cliente - qual é o melhor cartão para ele, e mostre este resultado para que ajude a você tomar uma decisão. A regra de predição seria achada usando a base acumulada de clientes do sistema.

Selecionando a Origem dos Dados

A janela **Origem de Dados** é similar à janela apresentada do OLAP vista anteriormente. Aqui você pode realizar as seguintes tarefas preparativas:

- Identificar as entidades, que serão usados para criar o modelo de predição;
- Os campos destas entidades serão carregados na lista **Campos**;

- Selecionar os campos que serão usados no modelo

Nota Você precisa sempre selecionar um campo que seja chave (primária) e que identifique os registros da tabela de casos que você vai pesquisar.

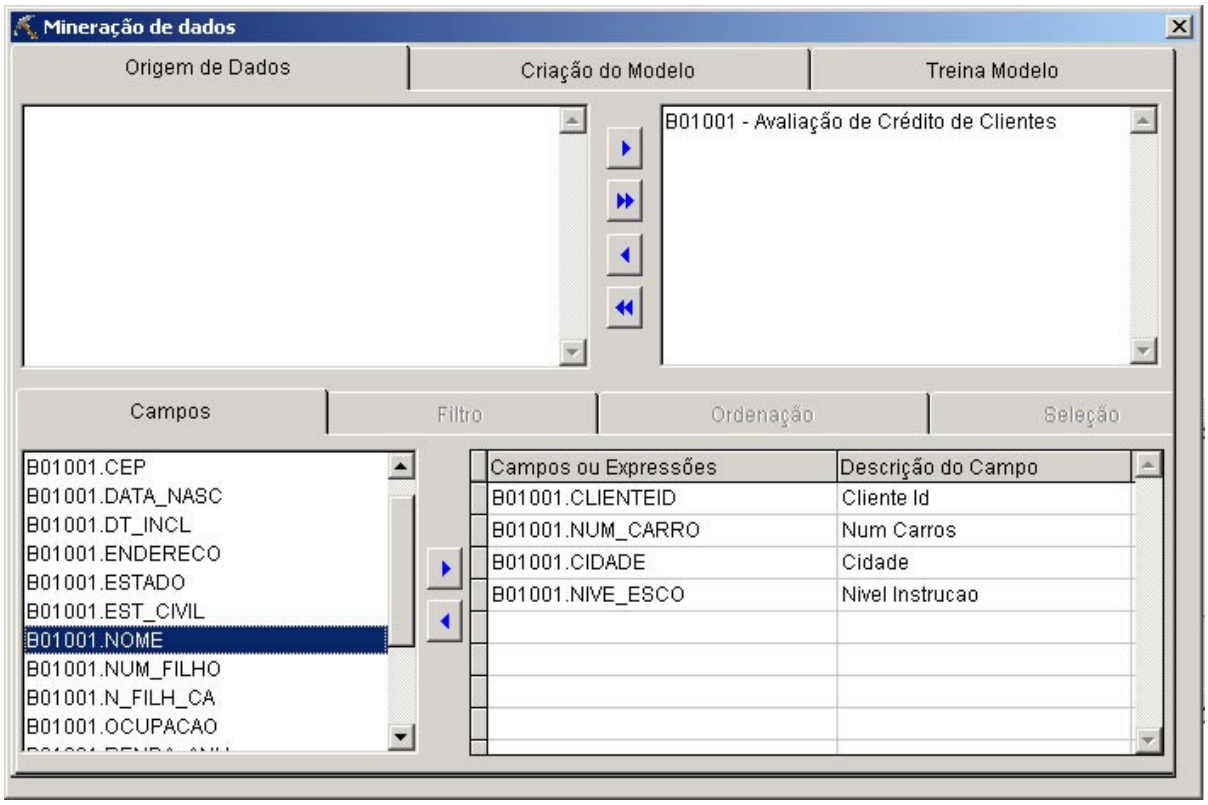


Fig 265 Seleção dos dados.

Criando o Modelo

Nesta página (veja na Fig 266) você indicará qual dos campos selecionado no passo anterior, vai ser o predito, ou seja, a saída do modelo, e quais serão usados para fazer a predição – as entrada do modelo. Os procedimentos a seguir são:

- Arraste primeiro o campo chave (que identificará cada caso) para o quadro “Chave”;
- Arraste o campo que será predito para o quadro “Predição”;
- Arraste os campos que farão parte da entrada do modelo, isto é, os campos a partir dos quais você irá predizer o modelo para o quadro Dados de Entrada;

- Na entrada de dados “Nome do Modelo”, preencha um nome para o modelo que você deseja criar e pressione o botão “Criar Modelo”
- Será criada a estrutura do modelo;

O modelo criado nesta fase realmente está “vazio”, ele precisa ser “treinado” com os dados da tabela de casos para poder predizer depois os eventos.

Campos de tipo hierárquicos (como por exemplo: cartão Platinum, Gold, Normal), tem que receber um peso, para poder avaliar bem a predição feita. Normalmente você deverá atribuir pesos para campos caracteres Fig 267. O peso tem que ser um número inteiro que indique a hierarquia.

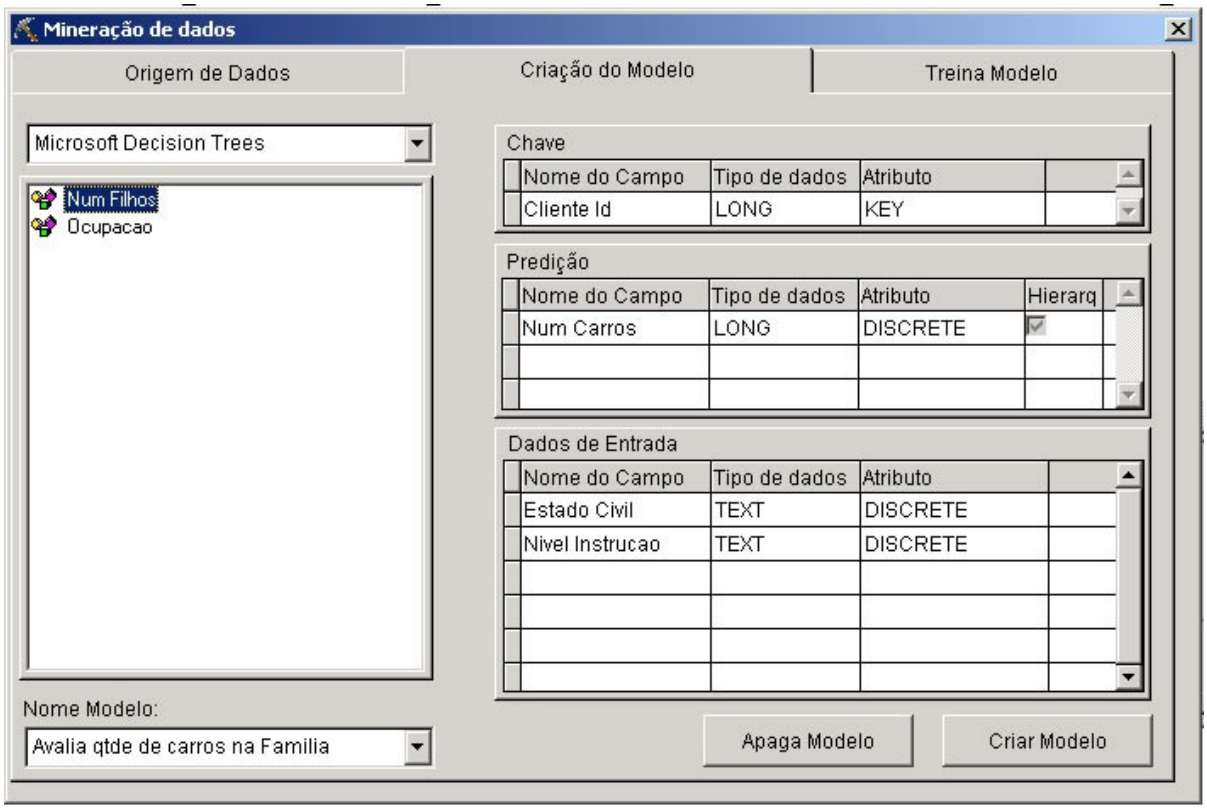


Fig 266 Criando o modelo.

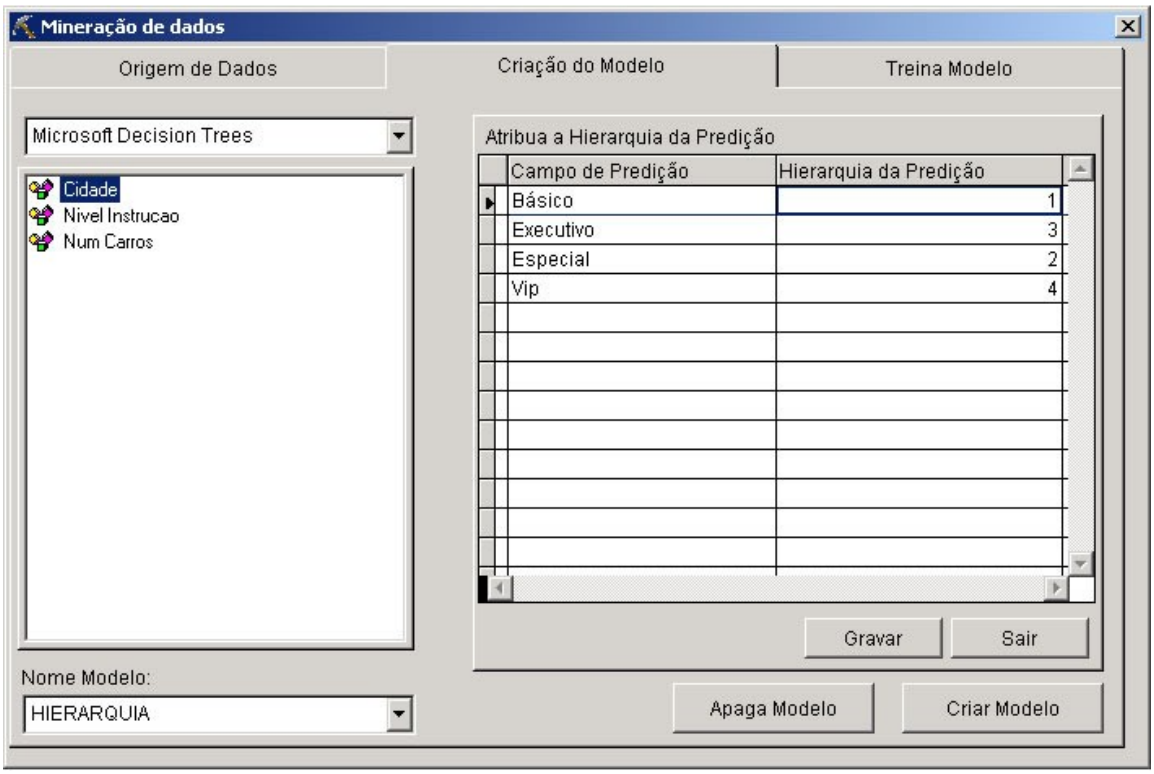


Fig 267 Colocando valores a campos hierárquicos.

Treinando e testando o modelo

Nesta fase (veja a janela na Fig 268) você faz o treinamento do modelo. O treinamento consiste em cálculos matemáticos sofisticados que são realizados para achar as regras de predição, mas que são totalmente transparentes para você. Os procedimento são:

- Selecione o modelo na entrada de dados.
- Na entrada de dados **Percentual de Treinamento**, selecione o percentual de registros (casos) que serão utilizados para o treinamento do modelo. Os registros que não forem utilizados no treinamento, serão utilizados para validar o modelo.

No caso da Fig 268, 80% da tabela de casos serão utilizados no treinamento do modelo e 20% serão utilizados para validar o modelo preceptivo.

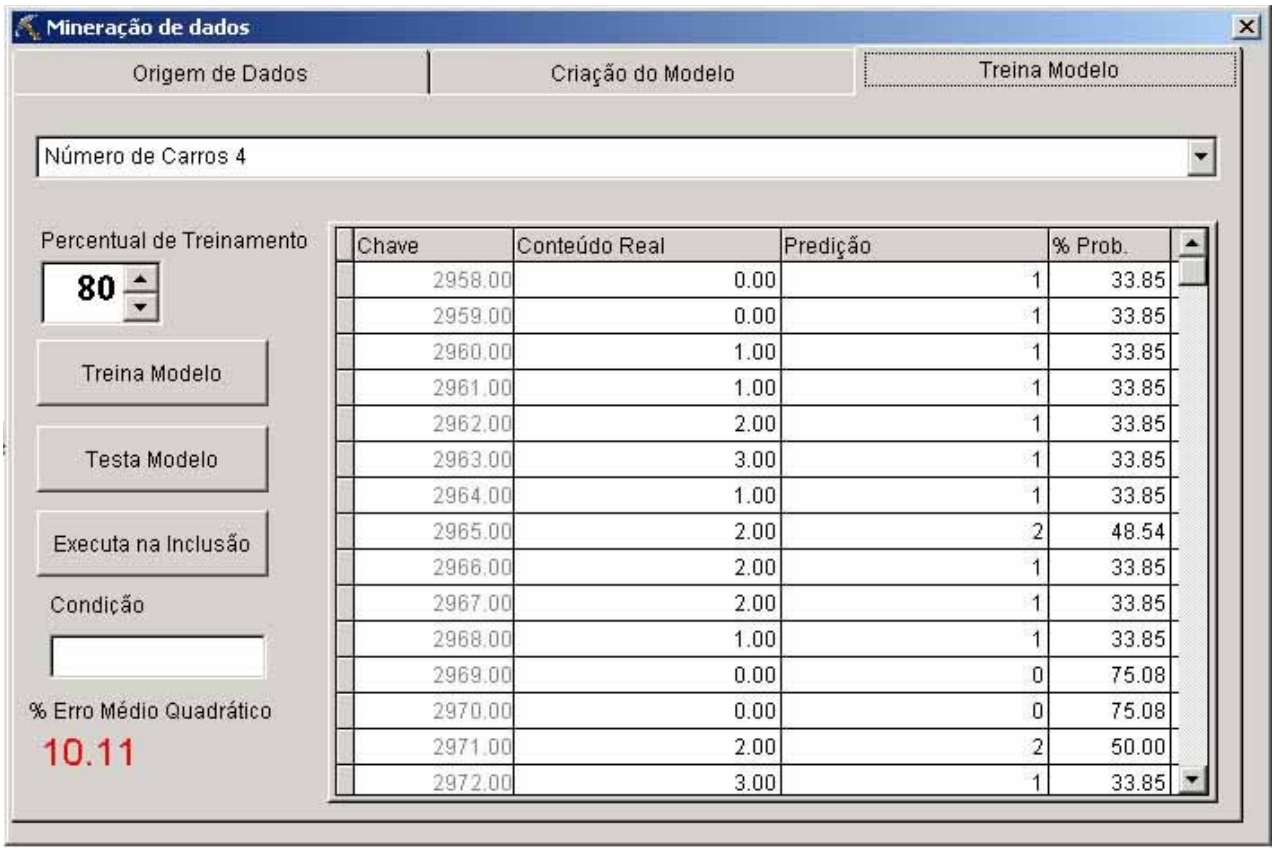


Fig 268 Treinando e Testando o modelo.

- Clique o botão “Treina Modelo” para treinar o modelo.

O modelo começará a ser treinado, às vezes pode levar um tempo, em dependência dos dados do modelo e o desempenho de seu computador. Depois de finalizado o treinamento é hora de avaliar o modelo.

A avaliação é feita clicando o botão “Testa Modelo”. Para o teste serão usados os casos que não participaram da criação das regras. Aqui serão comparando o valor real com o predito.

Ao final do teste o quadro será preenchido com: o campo chave da tabela de casos, o conteúdo real do campo, o valor predito e a probabilidade de predição de cada registro – ou seja, a probabilidade que o valor predito é certo.

Também é calculado o valor do **Erro Médio Quadrático** que é um parâmetro que mede de maneira geral o desempenho do modelo e que podemos usar para fazer comparações na escolha do melhor modelo. Quanto menor for o Erro Médio Quadrático, melhor será o modelo.

Usando o modelo

A última fase é embutir o modelo dentro do formulário para usá-lo. Clique no botão “Executa na Inclusão’ para atribuir a predição ao campo predito automaticamente após a inclusão de registros.

XVIII. Apêndices

A. Convertendo sistemas da versão 4.6 para a versão 6.0 do Visual Kit 5

A Ebendinger disponibiliza em seu site, um programa para converter sistemas gerados com a versão 4.6 do Visual Kit 5 para a nova versão do Visual Kit 5.

O programa conversor cria um novo diretório, converte o sistema para a nova versão e copia os arquivos para os novos sub-diretórios do sistema gerado.

Na cópia dos arquivos o conversor organiza o diretório do sistema da seguinte forma:
sub-diretórios no sistema convertido:

\SOURCE programas fontes (funções específicas)
\PARAMETERS\FORM formulários (scx)
\PARAMETERS\REPORT relatórios (frm)
\PARAMETERS\LABEL etiquetas (lbx)

os demais arquivos são organizados por tipo do dado armazenado.

B. Atualizando o Visual Kit 5 e sistemas gerados

A Ebendinger criou e disponibiliza em seu site, um programa para atualizar o Visual Kit 5 e os sistemas gerados.

Sempre que a Ebendinger concluir um ajuste ou incluir novidades no Visual Kit 5 ou nos sistemas gerados, os arquivos alterados serão disponibilizados em nosso servidor de FTP (*File Protocol Transfer*).

Faça o “download free” do programa de atualização, e execute o programa periodicamente para verificar se existem novidades em nosso servidor de FTP.

O download dos ajustes é permitido somente aos usuários cadastrados do Visual Kit 5. Veja na figura abaixo o funcionamento do programa de update (atualização).

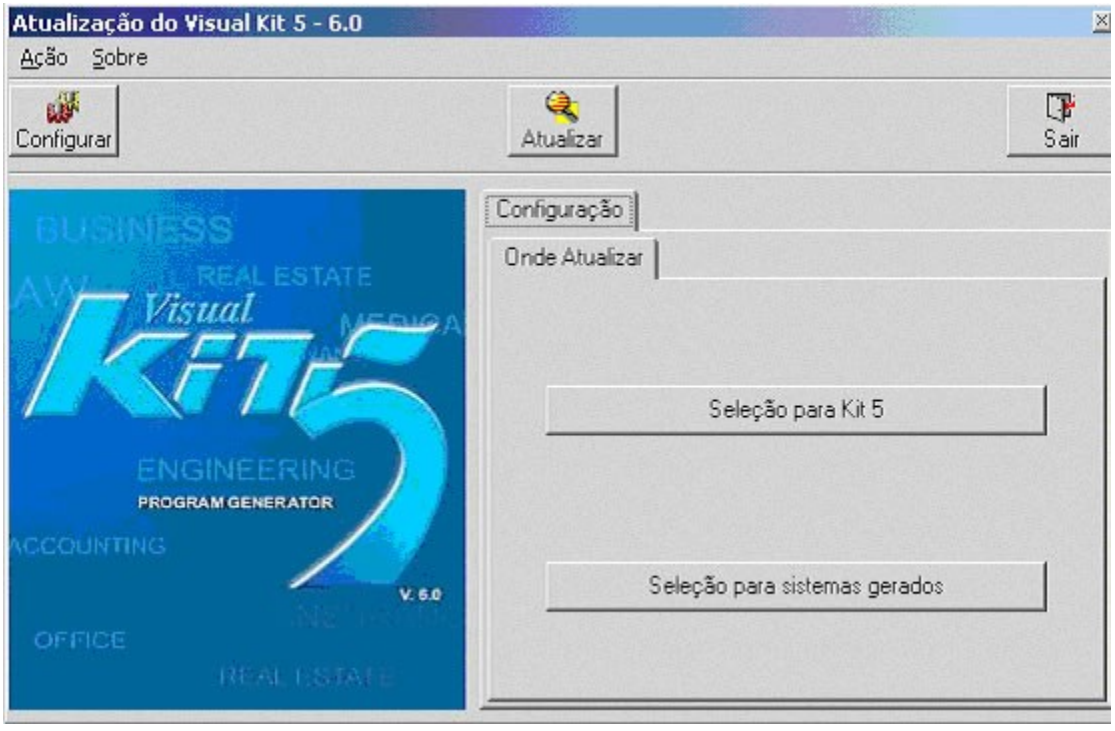


Fig 269 Programa de atualização do Visual Kit 5.

Pressione o botão "Seleção para Kit 5", para selecionar o diretório do Visual Kit 5 em seu micro, como no exemplo abaixo.

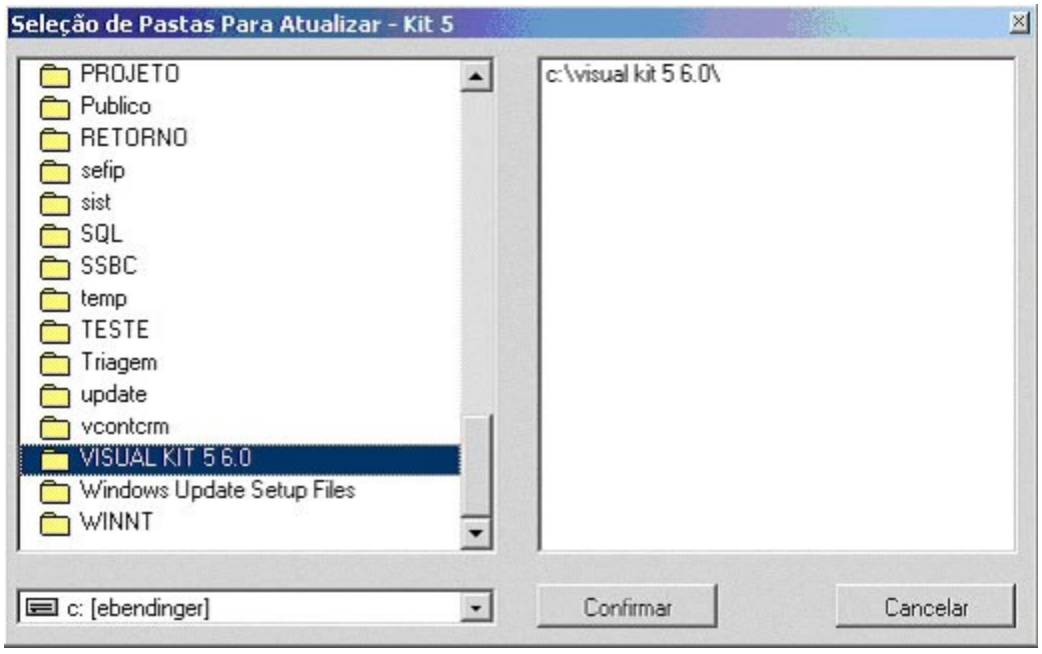


Fig 270 Selecione o diretório do Visual Kit 5

Pressione o botão "Seleção para sistemas gerados", para selecionar os diretórios dos sistemas gerados que você deseja atualizar.

Ao final da seleção de diretórios, pressione o botão "Atualizar", para iniciar o download e cópia dos arquivos para os diretórios selecionados.

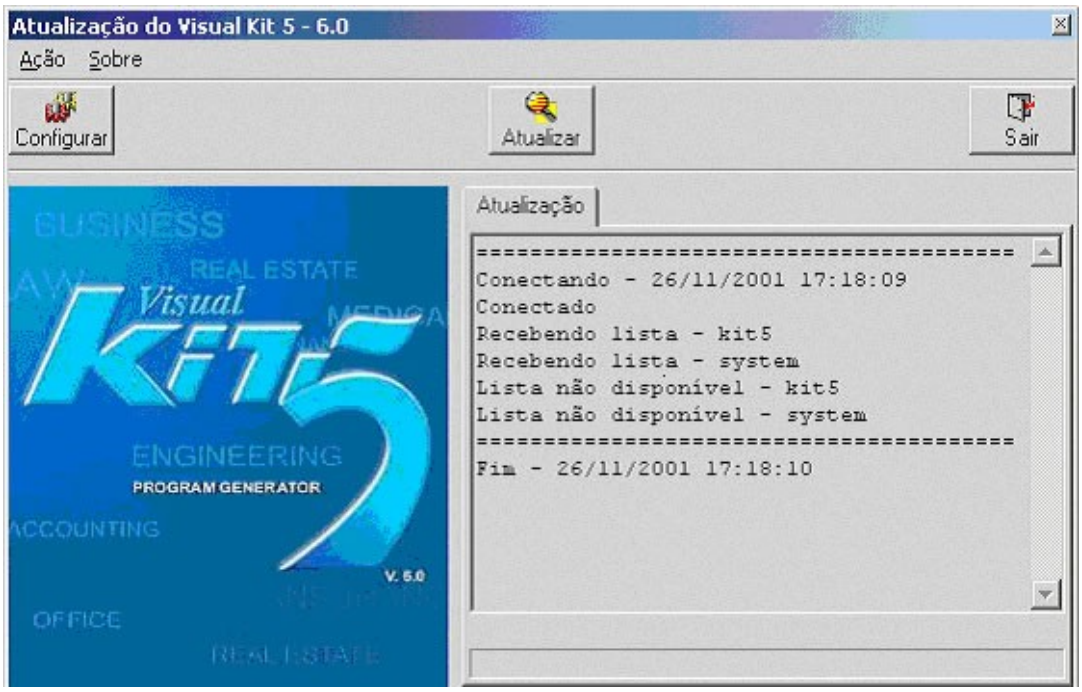


Fig 271 Fim da atualização.

C. Denominação das Tabelas, Módulos, Entidades no Visual Kit 5

No Visual Kit 5, as tabelas são enumeradas internamente com a letra "**B**" seguida com cinco dígitos cumprindo sempre o seguinte padrão:

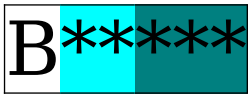


Fig 272 Denominação das tabelas no Visual Kit 5

em que:

- Os primeiros dois dígitos indicam o número do módulo a que pertence.
- Os restantes três dígitos indicam o número da entidade.

tendo em conta que os números dos módulos é atribuído pela ordem de criação dos mesmos. O primeiro módulo criado será o "**01**", o segundo "**02**" e assim por diante. O mesmo acontece com a enumeração das entidades.

Exemplo . B**01**001 – tabela da primeira entidade do primeiro módulo.

B**03**002 – tabela da segunda entidade do terceiro módulo

D. Propriedade Formatos e Máscara de Edição

Usando o controle *TextBox*, que é sem duvida a forma mais usada de inserir dados, o usuário pode inserir qualquer texto. Mas as vezes é preciso gerenciar essa entrada de dados para que o texto seja inserido de uma formatação ou apresentação especial, coisas como permitir a entrada somente de caracteres maiúsculos ou somente números ou até mesmo formatação específica de telefone ou CPF. Colocar estas regras é muito fácil de fazer usando as propriedades de Formato do Campo e Máscara de Edição.

Os formatos de dados são utilizados não só para exibir o conteúdo dos dados no monitor, mas também na usá-lo na impressão, por exemplo, você poderia imprimir somente uma parte dos dados. Veja na Fig 273 onde é localizada estas propriedades.

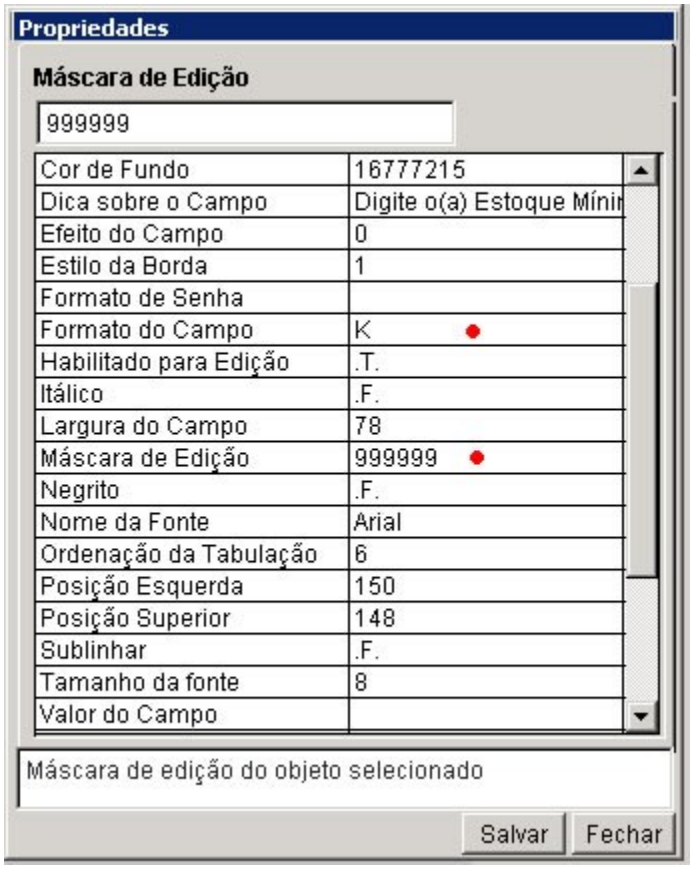


Fig 273 Localização das propriedades Formato do Campo e Mascara de Edição

O propriedade **Formato do Campo** possui vários valores que descrevemos na tabela:

Definição	Descrição
!	Converte para maiúsculas caracteres. Utiliza apenas para campos caractere.
$	Exibe o símbolo monetário. O campo deve ser numérico.
^	Exibe valores numéricos com notação científica.
A	Permite apenas caracteres alfabéticos (sem espaços ou marcas de pontuação).
D	Usa o formato de Data especificado.
E	Edita campo Data como data Britânicas.
K	Seleciona todo o texto quando o controle recebe o foco.
L	Exibe zeros à esquerda no lugar de espaços. O campo deve ser numérico.
R	Exibe a mascara do formato para o *TextBox* especificada na propriedade Máscara de Edição que veremos a continuação. Exemplo: se a máscara for 99-999, o número 12345 será exibido assim 12-345, mais o hifem não será gravado. Este somente para campo caractere ou numérico.
T	Retira os espaços para o campo digitado.
YS	Exibe campo data no formato resumido.
YL	Exibe campo data no formato longo, com hora.

A propriedade **Mascara de Edição** permite mascarar a entrada de dados realizada pelo usuário. Os caracteres definidos para a definição da máscara estão descritos na tabela XX

Máscara	Descrição
X	Permite a entrada de qualquer caractere.
9	Permite a digitação de dígitos e sinais, como o de subtração (–).

#	Dígitos, espaço em branco e sinais podem ser digitados.
$	Exibe o símbolo monetário numa posição fixa.
$$	Exibe o símbolo monetário em uma posição flutuante, próximo a qualquer digito num *Spinner* ou *TextBox*.
*	Asteriscos serão exibidos à esquerda do valor.
.	Um ponto especifica a posição do ponto decimal.
,	Virgulas podem ser incluídas para separar os dígitos a esquerda do ponto decimal.

Nota

Os sistemas gerados pelo Visual Kit 5 suportam os mesmos formatos (*InputMask* e *Format*) da linguagem Visual FoxPro v.6.0.

Exemplos:

Máscara para campo Data: ##/##/####.
Campo do tipo DATA.
Exemplo: 16/12/1964.

Tipo de Campo: *Numérico*
Tamanho do Campo: *10*
Decimais: *2*
Formato: *$*
Mascara: *9,999,999.99*
Exemplo: Salário, Valor, Preço.

Tipo de Campo: *Numérico*
Tamanho do Campo: *11*
Decimais: *0*
Formato: *K*
Mascara: *999-9999-9999*
Exemplo: Telefone, Fax.

E. Operadores

Os “Operadores”permitem que você manipule dados do mesmo tipo. Os operadores são agrupados da seguinte forma.

- Operadores de caracteres
- Operadores de data
- Operadores Lógicos
- Operadores Relacionais
- Operadores Numéricos

Operadores de Caracteres

Você pode juntar e comparar dados usando os operadores +, -, e $.

Operador	**Ação**	**Exemplo**
+	Concatenação. Junta duas seqüências de caracteres (*strings*).	'Bom ' + 'dia'
-	Concatenação. Elimina espaços em branco do elemento que precede ao operador, posteriormente junta os dois elementos	client.Nom - client.SobreNom
$	Comparação. Procura uma expressão caractere em outra	'father' $ 'grandfather'

Operadores de Datas

Operador	**Ação**	**Exemplo**
+	Adição	dNewDate = dDate1 + nDays
-	Subtração	dNewDate = dDate1 - nDays

Operadores Lógicos

Os operadores lógicos trabalham com qualquer tipo de dados e retorna um valor Lógico.

Operador	**Ação**	**Exemplo**
( )	Expressão de	Var1 AND (Var2 AND

	grupo	Var3)
NOT, !	Negativo Lógico	IF NOT VarA = VarB IF ! nVar1 = nVar2
AND	Lógico E	Campo0 AND Campo9
OR	Lógico OU	CampoX OR CampoY

Operadores Numéricos

A seguinte tabela mostra os operadores numéricos em ordem de precedência.

Operador	**Ação**	**Exemplo**
( )	Agrupa sub-expressão	(4-3) * (12/nVar2)
**, ^	Exponencial	3 ** 2 3 ^ 2
*, /	Multiplicação e divisão	2 * 7 14 / 7
%	Módulo (resto)	15 % 4
+, -	Adição e subtração	4 + 15

Operadores Relacionais

Os operadores relacionais trabalham com todos os tipos de dados e retorna um valor Lógico

Operador	**Ação**	**Exemplo**
<	Menos que	23 < 54
>	Maior que	1 > 2
=	Igual a	Var1 = Var2
<>, #, !=	Não igual a	.T. <> .F.
<=	Menor que ou igual a	1998/02/16 <= 1998/02/16
>=	Maior que ou igual a	32 >= 31
==	Comparação de caracteres string	status == "Aberto"

F. Regras de nomenclaturas Ebendinger

1. **Funções Específicas**
2. **Sub-funções**
3. **Funções de uso geral**
4. **Variáveis**
5. **Formulários**
6. **Etiquetas**
7. **Imagens**
8. **Help**
9. **Ícones**
10. **Documentos Word e outros**
11. **DataBase**
12. **Table (arquivo)**
13. **Índice**
14. **Query**
15. **View**
16. **Classe**
17. **Arquivo Texto**
18. **Menu**
19. **Report**

Funções Específicas:
São nomeadas pelo Visual Kit 5.

Exemplo: **E0301526**

Colunas	Descrição
E	Específica (função)
03	Terceiro Módulo
015	Décima quinta Entidade
2	Com abertura de arquivo pela APP
6	Etiqueta

Tipo de Função	Descrição
1	Função Básica da APP
2	Func. Específica com abertura de arquivos pela APP

	(bearq)
3	Sem abertura de arquivo. O programador pode abrir dentro da função se necessário.
4	Reservada
5	Chamada a executável externo

Tipo de Operação	Descrição
1	Inclusão
2	Alteração
3	Exclusão
4	Consulta
5	Relatório
6	Etiqueta
7	Reservada
8	Reservada
9	Transferência

Sub-funções:
Funções chamadas e criadas dentro de uma função específica. Nomeadas pelo programador da função específica. Usar o nome da função específica onde a sub-função foi originalmente criada, acrescido de um número seqüencial ascendente, iniciado em 001 até 999.

Exemplo: **S0301526020**()

Colunas	Descrição
S	Sub-função
03	Criada originalmente no módulo 03
015	Criada originalmente para a entidade 015
2	Com abertura de arquivo. Repete o byte usado nesta coluna no nome da função específica Primitiva (original)
6	Etiqueta. Repete o byte usado nesta coluna no nome da função específica Primitiva.
020	Sub-função de número 020, na função específica E0301526. Número seqüencial ascendente.

Diretório: Todas as funções específicas e sub-funções devem ser gravadas no sub-diretório *SOURCE\D01...D02.. D99* de acordo com o módulo ao qual pertence a entidade, no diretório do sistema gerado.

Exemplo: **H:\CREMERJ\SOURCE\d03**

Funções de uso geral:
Criadas para utilização por mais de uma função específica ou sub-função. Nomeadas pelo programador da função. Usar um nome que permita uma fácil identificação.

Os dois primeiros bytes do nome da função indicam o valor de retorno da função.

Exemplo: **lgMontaFiltro()**

Tipos de Função (por retorno)	Descrição do valor de Retorno
Ct	Caracter
Dt	Data
Nm	Numérico
Cn	Currency (moeda)
Fl	Float
Dm	DateTime
Db	Double
In	Inteiro
Lg	Lógico
Mm	Memo
Gn	General (imagem, ole...)

Diretório: Todas as funções de uso geral devem ser gravadas no sub-diretório *SOURCE\general*, no diretório do sistema gerado.

Exemplo: **H:\CREMERJ\SOURCE\general**

Variáveis:
Criadas nas funções específicas, sub-funções e funções de uso geral. Nomeadas pelo programador da função. Usar um nome que permita uma fácil identificação.

Os dois primeiros bytes do nome da variável indicam o tipo de dado da variável.

Usar sempre que possível palavras completas, iniciadas por letra maiúscula.

Vide tabela acima (funções de uso geral).

Exemplo: **ctNomeCliente**

Formulários:
Formulários criados no Visual FoxPro. Arquivos SCX. Usar o nome da função específica onde o formulário foi originalmente criado, acrescido de um número sequencial ascendente, iniciado em 01 até 99.

Exemplo: **F030152501.SCX**

Colunas	Descrição
F	Formulário
03	Criado originalmente no módulo 03. Repete os bytes usados nesta coluna no nome da função específica Primitiva.
015	Criado originalmente para a entidade 015. Repete os bytes usados nesta coluna no nome da função específica Primitiva.
2	Com abertura de arquivo. Repete o byte usado nesta coluna no nome da função específica Primitiva.
5	Relatório. Repete o byte usado nesta coluna no nome da função específica Primitiva.
01	Formulário de número 01.

Diretório: Todos os formulários devem ser gravados no sub-diretório *PARAMETERS\FORM\D01, D02... D99* de acordo com o módulo ao qual pertence a entidade, no diretório do sistema gerado.

Exemplo: **H:\CREMERJ\parameters\form\d03**

Etiquetas:
Etiquetas (label) criadas no Visual FoxPro. Arquivos LBX. Usar o nome da função específica onde a etiqueta foi originalmente criada, acrescido de um número seqüencial ascendente, iniciado em 01 até 99.

Exemplo: **L030152602.LBX**

Colunas	Descrição

L	Label (etiqueta)
03	Criado originalmente no módulo 03. Repete os bytes usados nesta coluna no nome da função específica Primitiva.
015	Criado originalmente para a entidade 015. Repete os bytes usados nesta coluna no nome da função específica Primitiva.
2	Com abertura de arquivo. Repete o byte usado nesta coluna no nome da função específica Primitiva.
6	Etiqueta. Repete o byte usado nesta coluna no nome da função específica Primitiva.
02	Etiqueta de número 02.

Diretório: Todas as etiquetas devem ser gravadas no sub-diretório *PARAMETERS\LABEL\D01, D02... D99* de acordo com o módulo ao qual pertence a entidade, no diretório do sistema gerado.

Exemplo: **H:\CREMERJ\parameters\label\d03**

Imagens:

Imagens usadas no sistema. Arquivos BMP, GIF, JPG. Usar o nome da função específica para a qual a imagem foi originalmente criada, acrescido de um número sequencial ascendente, iniciado em 001 até 999.

Exemplo: **I0301525002.GIF**

Colunas	**Descrição**
I	Imagem
03	Criada originalmente no módulo 03. Repete os bytes usados nesta coluna no nome da função específica Primitiva.
015	Criada originalmente para a entidade 015. Repete os bytes usados nesta coluna no nome da função específica Primitiva.
2	Repete o byte usado nesta coluna no nome da função específica Primitiva.
5	Relatório. Repete o byte usado nesta coluna no nome da função específica Primitiva.
002	Imagem de número 002.

Diretório: Todas as imagens devem ser gravadas no sub-diretório *PARAMETERS\IMAGE\D01, D02.. D99* de acordo com o módulo ao qual pertence a entidade, no diretório do sistema gerado.

Exemplo: **H:\CREMERJ\parameters\image\d03**

Help:
Arquivos de ajuda do sistema gerado, criados no Word, compilados no Visual FoxPro. Arquivos HLP.

Diretório: Todos os arquivos de help devem ser gravados no sub-diretório *PARAMETERS\HELP*, no diretório do sistema gerado.

Ícones:
Arquivos criados usando programas gráficos. Arquivos ICO.

Diretório: Todos os arquivos de ícone devem ser gravados no sub-diretório *PARAMETERS\ICO*, no diretório do sistema gerado.

Documentos Word e outros:
Os arquivos são nomeados pelo funcionário que os criar.

Exemplo: **D990518011.DOC**

Colunas	Descrição
D	Letra padrão.
99	Ano de criação do documento. (98, 99...)
05	Mês de criação do documento. (01,02..12)
18	Dia de criação do documento (01,02...31)
011	Número sequencial de 001 à 999, diarios. Por cliente.

Diretório: Todos os arquivos Word ou de outros softwares (excel, powerpoint...) devem ser gravados no sub-diretório Editor/[nome do cliente], no driver de rede designado.

Não é permitido manter documentos nos drivers das Estações.

Dentro do arquivo deve ser gravado o nome do arquivo, para facilitar a busca.

DataBase:

DataBase (Base de Dados) criadas no Visual FoxPro, Access etc. Usar o nome do sistema para nomear o DataBase.

Exemplo: **CREMERJ.DBC**

Diretório: Todos os DataBase devem ser gravados no sub-diretório *DATABASE*, no diretório do sistema gerado.

Exemplo: **H:\CREMERJ\database**

Table (arquivo):
Tabelas (arquivos) criadas no Visual FoxPro. Arquivos DBF. Sequir as convenções do Visual Kit 5 para nomear tabelas.

As tabelas para uso geral devem ser nomeadas de traz pra frente.

Exemplo: **B04999.DBF**

Colunas	Descrição
B	Padrão interno.
04	Criada originalmente no módulo 04.
999	Criada originalmente para a entidade 999. As tabelas de uso geral devem ser criadas com a numeração descendente.

Diretório: Todas as tabelas devem ser gravadas no sub-diretório do módulo correspondente *D04\DBF*, no diretório do sistema gerado.

Exemplo: **H:\CREMERJ\D04\DBF**

Índice (indexador):
Arquivos de Índice criados no sistema gerado ou Visual Kit 5. Arquivos IDX. Sequir as convenções do Visual Kit 5 para nomear índices.

Os arquivos de índice devem ser idênticos ao arquivo de dados (dbf), acrescidos de um número sequencial de 1 a 14.

Exemplo: **B020013.IDX**

Colunas	Descrição
B	Padrão interno.
02	Criado originalmente no módulo 02.

001	Criado originalmente para a entidade 001. As tabelas de uso geral devem ser criadas com a numeração descendente.
3	Terceiro arquivo de índice fixo.

O limite de arquivos de índice fixos é 14.

Os arquivos de índices fixos devem ser especificados na tabela de parâmetros (bearq) “Abertura de Arquivos” do Visual Kit 5.

IMPORTANTE: Todas as funções específicas que atualizam dados em tabelas do sistema gerado (inclusão, alteração ou exclusão de dados), devem abrir no (bearq) “Abertura de Arquivos” do Visual Kit 5, todos os índices fixos da tabela atualizada.

Exemplo: arquivo B02001.DBF
Índices fixos: BO20011.IDX
B020012.IDX
B020013.IDX

Se uma função específica atualiza dados no B02001.DBF, então você deverá cadastrar no “Abertura de Arquivos” do Visual Kit 5, todos estes índices para esta função específica com abertura de arquivo.

Diretório: Todos os arquivos de índice devem ser gravadas no sub-diretório do módulo correspondente *D04\IDX*, no diretório do sistema gerado.

Exemplo: **H:\CREMERJ\D04\IDX**

Query:
Pesquisa pré-programada (query) criadas no Visual FoxPro. Arquivos QPR. Usar o nome da função específica para a qual a *query* foi originalmente criada, acrescido de um número seqüencial ascendente, iniciado em 01 até 99.

Exemplo: **Q030152601.QPR**

Colunas	Descrição
Q	Query (pesquisa)

03	Criada originalmente no módulo 03. Repete os bytes usados nesta coluna no nome da função específica Primitiva.
015	Criada originalmente para a entidade 015. Repete os bytes usados nesta coluna no nome da função específica Primitiva.
2	Com abertura de arquivo. Repete o byte usado nesta coluna no nome da função específica Primitiva.
6	Etiqueta. Repete o byte usado nesta coluna no nome da função específica Primitiva.
01	Query de número 01.

Diretório: Todas as querys devem ser gravadas no sub-diretório *PARAMETERS\QUERY\D01, D02... D99* de acordo com o módulo ao qual pertence a entidade, no diretório do sistema gerado.

Exemplo: **H:\CREMERJ\parameters\query\d03**

View:

Visões do arquivo (view) criadas no Visual FoxPro. Arquivos VUE. Usar o nome da função específica onde a view foi originalmente criada, acrescido de um número seqüencial ascendente, iniciado em 01 até 99.

Exemplo: **V030152602.VUE**

Colunas	**Descrição**
V	View (visão do arquivo)
03	Criada originalmente no módulo 03. Repete os bytes usados nesta coluna no nome da função específica Primitiva.
015	Criada originalmente para a entidade 015. Repete os bytes usados nesta coluna no nome da função específica Primitiva.
2	Com abertura de arquivo. Repete o byte usado nesta coluna no nome da função específica Primitiva.
6	Etiqueta. Repete o byte usado nesta coluna no nome da função específica Primitiva.
02	View de número 02.

Diretório: Todas as visões do arquivo devem ser gravadas no sub-diretório *PARAMETERS\VIEW\D01, D02... D99* de acordo com o módulo ao qual pertence a entidade, no diretório do sistema gerado.

Exemplo: **H:\CREMERJ\parameters\view\d03**

Classe:
Classes criadas no Visual FoxPro. Arquivos VCX. Usar o nome da função específica onde a classe foi originalmente criada, acrescido de um número sequencial ascendente, iniciado em 01 até 99.

Exemplo: **C030152603.VCX**

Colunas	Descrição
C	Classe
03	Criada originalmente no módulo 03. Repete os bytes usados nesta coluna no nome da função específica Primitiva.
015	Criada originalmente para a entidade 015. Repete os bytes usados nesta coluna no nome da função específica Primitiva.
2	Com abertura de arquivo. Repete o byte usado nesta coluna no nome da função específica Primitiva.
6	Etiqueta. Repete o byte usado nesta coluna no nome da função específica Primitiva.
03	Classe de número 03.

Diretório: Todas as classes devem ser gravadas no sub-diretório *PARAMETERS\CLASS\D01, D02... D99* de acordo com o módulo ao qual pertence a entidade, no diretório do sistema gerado.

Exemplo: **H:\CREMERJ\parameters\class\d03**

Arquivo Texto:
Criados no Visual FoxPro, Sistema gerado ou Visual Kit 5. Arquivos TXT. Usar o nome da função específica onde o arquivo texto foi originalmente criado, acrescido de um número seqüencial ascendente, iniciado em 001 até 999.

Exemplo: **T0301526002.TXT**

Colunas	Descrição
T	Texto
03	Criado originalmente no módulo 03. Repete os bytes usados nesta coluna no nome da função específica Primitiva.
015	Criado originalmente para a entidade 015. Repete os bytes

	usados nesta coluna no nome da função específica Primitiva.
2	Com abertura de arquivo. Repete o byte usado nesta coluna no nome da função específica Primitiva.
6	Etiqueta. Repete o byte usado nesta coluna no nome da função específica Primitiva.
002	Arquivo Texto de número 002.

Diretório: Todos os arquivos de texto devem ser gravados no sub-diretório *PARAMETERS\TEXTFILE\D01, D02... D99* de acordo com o módulo ao qual pertence a entidade, no diretório do sistema gerado.

Exemplo: **H:\CREMERJ\parameters\textfile\d03**

Menu:

Menus criados no Visual FoxPro. Arquivos MNX. Usar o nome da função específica onde o menu foi originalmente criado, acrescido de um número seqüencial ascendente, iniciado em 01 até 99.

Exemplo: **M030152602.MNX**

Colunas	Descrição
M	Menu
03	Criado originalmente no módulo 03. Repete os bytes usados nesta coluna no nome da função específica Primitiva.
015	Criado originalmente para a entidade 015. Repete os bytes usados nesta coluna no nome da função específica Primitiva.
2	Com abertura de arquivo. Repete o byte usado nesta coluna no nome da função específica Primitiva.
6	Etiqueta. Repete o byte usado nesta coluna no nome da função específica Primitiva.
02	Menu de número 02.

Diretório: Todos os menus devem ser gravados no sub-diretório *PARAMETERS\MENU\D01, D02... D99* de acordo com o módulo ao qual pertence a entidade, no diretório do sistema gerado.

Exemplo: **H:\CREMERJ\parameters\menu\d03**

Report:

Relatórios criados no Visual FoxPro. Arquivos FRX. Usar o nome da função específica onde o relatório foi originalmente criado, acrescido de um número seqüencial ascendente, iniciado em 01 até 99.

Exemplo: **R030152501.FRX**

Colunas	Descrição
R	Relatório (Report)
03	Criado originalmente no módulo 03. Repete os bytes usados nesta coluna no nome da função específica Primitiva.
015	Criado originalmente para a entidade 015. Repete os bytes usados nesta coluna no nome da função específica Primitiva.
2	Com abertura de arquivo. Repete o byte usado nesta coluna no nome da função específica Primitiva.
5	Relatório. Repete o byte usado nesta coluna no nome da função específica Primitiva.
01	Relatório de número 01.

Diretório: Todos os relatórios devem ser gravados no sub-diretório *PARAMETERS\REPORT\D01, D02... D99* de acordo com o módulo ao qual pertence a entidade, no diretório do sistema gerado.

Exemplo: **H:\CREMERJ\parameters\report\d03**